CRAY T3D™ Administrator's Guide

SG–2507 1.1

Cray Research, Inc.

Portions of this product may still be in development. The existence of those portions still in development is not a commitment of actual release or support by Cray Research, Inc. Cray Research, Inc. assumes no liability for any damages resulting from attempts to use any functionality or documentation not officially released and supported. If it is released, the final form and the time of official release and start of support is at the discretion of Cray Research, Inc.

Requests for copies of Cray Research, Inc. publications should be sent to the following address:

Cray Research, Inc.
Distribution Center
2360 Pilot Knob Road
Mendota Heights, MN 55120
USA

Order desk (612) 683–5907
Fax number (612) 452–0141

Cray Research Software Documentation Map

The illustration on the following pages highlights the major body of documentation available for Cray Research (CRI) customers. The illustration is organized into categories by audience designation:

Audience	Description
End users	Those who use the UNICOS operating system, products, applications, or linking software
Application and system programmers	Those who write or modify system or application code on a CRI system for the purpose of solving computer system, scientific, or engineering problems
System administrators	Those who perform system administration tasks, such as installation, configuration, and basic troubleshooting
System analysts	Those who perform advanced troubleshooting, tuning, and customization
Operators	Those who perform operational functions, such as performing system dumps, and those who administer an operator workstation

To use the map, find the audience designation closest to your specific needs or role as a CRI system user. Note that manuals under other audiences may also be of interest to you; manuals are listed only once, underneath the audience to which they most directly apply. Some manual titles are abbreviated. The date in the footer tells you when the information was last revised.

For more information

In addition to the illustration, you can use the following publications to find documentation specific to your needs:

- *Software Documentation Ready Reference,* publication SQ–2122, serves as a general index to the CRI documentation set. The booklet lists documents and man pages according to topic.
- *Software Overview for Users*, publication SG–2052, introduces the UNICOS operating system, its features, and its related products. It directs you to documentation containing user-level information.
- *User Publications Catalog*, publication CP–0099, briefly describes all CRI manuals available to you, including some not shown on the map, such as training workbooks and other supplementary documentation.

Ordering

To obtain CRI publications, order them by publication number from the Distribution Center:

Cray Research, Inc.
Distribution Center
2360 Pilot Knob Road
Mendota Heights, MN 55120
USA

Order desk (612) 683-5907
Fax number (612) 452-0141

END USERS

Introductory
Software Overview for Users (SG–2052)*

User's Guide to Online Information (SG–2143)*

General
Software Documentation Ready Reference (SQ–2122)*

User Commands Reference (SR–2011)†

User Commands Ready Reference (SQ–2056)†

Korn Shell Ready Reference (SQ–2115)

UNICOS Shells Ready Reference (SQ–2116)

UNICOS Environment Variables Ready Reference (SQ–2117)

UNICOS Index for Man Pages (SR–2049)

Visual Interfaces Guide (SG–3094)*

Tape Subsystem Guide (SG–2051)*

Security (MLS) Guide (SG–2111)

MPP Guide (SG–2508)*

CRL
CRL User's Guide (SG–2126)*

Networking
NQS Guide (SG–2105)*

TCP/IP and OSI Network Guide (SG–2009)*

Text Editing
Text Editors Primer (SG–2050)

`vi` Reference Card (SQ–2054)

`ed` Reference Card (SQ–2055)

MVS Link
MVS Station Messages (SI–0108)

Station Reference (SI–2066)

Station Ready Reference (SI–0104)

RQS User's Guide (SG–2405)

NOS/VE Link
NOS/VE Reference (SC–0270)

UNIX Link
RQS User's Guide (SG–2119)

VAX/VMS Link
SUPERLINK User's Guide (SV–3153)

RQS User's Guide (SV–3151)

Station Primer (SV–0361)

Station Reference (SV–0020)

Station Ready Reference (SV–0102)

VM Link
RQS VM User's Guide (SI–0170)

Station Primer (SI–0167)

Station Reference (SI–0168)

Station Messages and Codes (SI–0165)

Station Reference Summary (SI–0169)

* Available online with Docview

† Man pages available with the `man` command

APPLICATION AND SYSTEM PROGRAMMERS

Ada
Cray Ada Reference (SR–3014)

Cray Ada Programming Guide (SR–3082)

C
Cray Standard C Reference (SR–2074)*

Cray Standard C Ready Reference (SQ–2076)

Cray Standard C for MPP (SR–2506)*

CAL for CRAY Y-MP and CRAY Y-MP C90
Reference (SR–3108)

Symbolic Machine Instructions (SR–3109)

Ready Reference (SQ–3110)

UNICOS Macros and Opdefs (SR–2403)

Cray Assembler for MPP
CAM Reference (SR–2510)*

FORTRAN 77
CF77 Ready Reference (SQ–3770)

CF77 Commands and Directives (SG–3771)*

CF77 Fortran Reference (SR–3772)*

CF77 Optimization Guide (SG–3773)*

CF77 Message Manual (SR–3774)

Cray MPP Fortran Reference (SR–2504)*

Fortran 90
CF90 Commands and Directives (SR–3901)*

CF90 Fortran Language Reference (SR–3902)*

CF90 Ready Reference (SQ–3900)

Libraries
Fortran Library (SR–2079)†

Fortran Library Ready Ref. (SQ–2145)†

C Library (SR–2080)†

C Library Ready Ref. (SQ–2147)†

Scientific Libraries (SR–2081)†

Math Library (SR–2138)†

I/O User's Guide (SG–3075)*

Advanced I/O Guide (SG–3076)*

PVM and HeNCE Ref. (SR–2501)*

PVM Reference Card (SQ–2512)

Loaders
Loader Reference (SR–0066)*

SEGLDR Ready Reference (SQ–0303)

Networking
RPC Reference (SR–2089)*

Kerberos User's Guide (SG–2409)*

Programming Tools
Performance Utilities Reference (SR–2040)*

UNICOS Message System Programmer's Guide (SG–2121)*

Compiler Information File (CIF) Reference (SR–2401)*

CDBX Debugger Reference (SR–2091)*

CDBX Debugger User's Guide (SG–2094)

CDBX Reference Card (SQ–2110)

Program Browser (`xbrowse`) (IN–2140)

MPP Apprentice Tool (IN–2511)*

TotalView Debugger Ref. (SR–2502)*

Source Control
USM User's Guide (SG–2097)*

System Calls
System Calls (SR–2012)†

X Window System
Reference (SR–2101)*

Ready Reference (SQ–2123)

OPERATORS

OWS-E/IOS-E
OWS-E/IOS-E Reference (SR–3077)†

OWS-E/IOS-E Ready Reference (SQ–3080)

OWS-E/IOS-E Operator's Guide (SG–3078)

OWS-E/IOS-E Administrator's Guide (SG–3079)

Linking Software
CLS-UX (SU–3122)

MVS Station (SI–0037)

* Available online with Docview

† Man pages available with the `man` command

SYSTEM ADMINISTRATORS AND ANALYSTS

UNICOS

UNICOS Installation Guide (SG–2112)

Installation Ref. Card (SQ–2411)

UNICOS Installation Tool Menus and Help Files (SG–2412)

UNICOS System Administration (SG–2113)*

Administrator Commands Reference (SR–2022)†

Administrator Commands Ready Ref. (SQ–2413)†

CRL

CRL Administrator's Guide (SG–2127)*

DMF

DMF Administrator's Guide (SG–2135)*

Security

UNICOS System Security Overview (SG–2141)*

C2 Functionality on MLS Systems (SN–2407)

Networking

fy Driver Administrator's Guide (SG–2132)

OSI Administrator's Guide (SG–2142)

OSI

OSI Administrator's Guide (SG–2142)

MPP

CRAY T3D Administrator's Guide (SG–2507)

CRAY EL Series

CRAY Y-MP EL-specific Administrator Commands Reference (SR–2408)†

UNICOS Installation Guide for CRAY Y-MP EL Systems (SG–5201)

CRAY EL Series IOS Messages (SQ–2402)

VM Link

Station Installation and Maintenance (SI–0162)

SUPERLINK Administrator's Guide (SI–0171)

MVS Link

Station Installation (SI–0078)

RQS Administrator's Guide (SG–2406)

VAX/VMS Link

Station Installation (SV–0100)

Station Administration (SV–0363)

RQS Administrator's Guide (SV–3152)

SUPERLINK Administrator's Guide (SV–3154)

UNIX Link

RQS Administrator's Guide (SG–2120)

NOS/VE Link

NOS/VE Operator and Administrator Guide (SC–0271)

Analysts

File Formats and Special Files Reference (SR–2014)†

Data Migration MSP Writer's Guide (SN–2098)*

UNICOS Tuning Guide (SR–2099)

UNICOS `nmake` Card (SQ–2146)

Installation and Configuration Tool Reference (SR–3090)

USCP

Front-end Protocol Internals (SM–0042)*

USCP Optimization (SN–2103)

* Available online with Docview

† Man pages available with the `man` command

New Features

CRAY T3D Administrator's Guide SG–2507 1.1

The *CRAY T3D Administrator's Guide*, publication SG–2507 1.1, incorporates the following changes for the UNICOS MAX 1.1 release.

Two new sections were added:

- "CRAY T3D Configuration Planning" describes some considerations that should be taken into account when planning a CRAY T3D system initial configuration or later reconfiguration.
- "CRAY T3D System Messages" documents messages issued by CRAY T3D system software, along with an explanation, a severity level, and any action needed.

The following man pages were added:

- `blt_copy`(2)
- `olnx`(8)
- `olperi`(8)

The following concepts were documented:

- Express processing, which allows small jobs to initiate ahead of large jobs.
- Administrative resource pool shapes

Record of Revision

The date of printing or software version number is indicated in the footer. Changes in rewrites are noted by revision bars along the margin of the page.

Version	***Description***
1.0	December 1993. Original printing. Documentation to support UNICOS MAX release 1.0 running on Cray Research computer systems.
1.1	June 1994. Documentation to support UNICOS MAX release 1.1 running on Cray Research computer systems.

Preface

This publication documents system administration of a CRAY T3D system running the UNICOS MAX 1.1 operating system.

This publication provides both a conceptual overview of the responsibilities of the system administrator of a CRAY T3D system and a guide to the administrative tasks for maintaining, monitoring, and troubleshooting a CRAY T3D system.

This publication is written for CRAY T3D system administrators and analysts. CRAY T3D users should refer to the *Cray Research MPP Software Guide*, publication SG–2508.

This publication assumes that the reader is knowledgeable about system administration of a CRAY Y-MP system and the UNICOS operating system.

Related publications

The following related publications also will be useful in administering a CRAY T3D system:

Publication	Title
HR–04033	*CRAY T3D System Architecture Overview*
SG–2508	*Cray Research MPP Software Guide*
SG–5216	*UNICOS MAX Installation Guide*
SG–5217	*MPP Programming Environment Installation Guide*

Conventions

The following conventions are used throughout this manual:

Convention	Meaning
`Courier`	This font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
italic	This typeface denotes variable entries and words or concepts being defined.
`bold Courier`	This font denotes literal items that the user enters in screen drawings of interactive sessions. Output is shown in nonbold Courier font.
[ ]	Brackets enclose optional portions of a command line.
...	Ellipses indicate that a preceding command-line parameter can be repeated.

The following machine naming conventions are used throughout this manual:

Term	Definition
CRAY Y-MP systems	All configurations of CRAY Y-MP systems supported by UNICOS 8.0, including the M90 series (M92, M94, M98); C90 series (C916, C92A, C94, C94A, and C98); E series (2E, 4E, 8E, and 8I); EL series (including CRAY Y-MP EL, CRAY EL92, and CRAY EL98).
Cray MPP systems	All configurations of the CRAY T3D series, supported by UNICOS 8.0, including CRAY T3D MC, CRAY T3D MCA, and CRAY T3D SC.
All Cray Research systems	All configurations supported by UNICOS 8.0.

In this publication, *Cray Research*, *CRI*, and *Cray* refer to Cray Research, Inc. and/or its products.

Man page references

Throughout this document, reference is made to the online man pages available under UNICOS through the `man` command. A *man page* is a discussion of a particular element of the UNICOS operating system or a compatible product.

Each man page includes a general description of one or more commands, routines, system calls, or other topics, and provides details of their usage (command syntax, routine parameters, system call arguments, and so on). If more than one topic appears on a page, the entry in the printed manual is alphabetized under its primary name; online, secondary entry names are linked to these primary names. For example, `rc` is a secondary entry on the page with a primary entry name of `brc`. To access `rc` online, you can type `man rc`. To access information about `brc` online, you can type either `man rc` or `man brc`; both commands display the `brc` man page on your terminal.

Section numbers appear in parentheses after man page names. Man pages are referenced in text by entry name and section number, as shown in the following example:

> The `-p` and `-s` options to the `dmput(1)` command require that the caller be super user.

The following lists the type of entry associated with each section number:

Section	Subject
1	User commands
1B	User commands ported from BSD
2	System calls
3	Library routines, macros, and opdefs
4	Devices (special files)
4P	Protocols
5	File formats
7	Miscellaneous topics
7D	DWB-related information
8	Administrator commands

A routine name followed by an empty set of parentheses designates a kernel routine; for example, `ddcntl()`. These routines do not have man pages associated with them.

Printed man pages are published in Cray Research manuals. The following manuals consist of collections of man pages that describe the UNICOS operating system commands, system calls, and file formats:

Publication	Title
SR–2011	*UNICOS User Commands Reference Manual*
SR–2012	*UNICOS System Calls Reference Manual*
SR–2014	*UNICOS File Formats and Special Files Reference Manual*
SR–2022	*UNICOS Administrator Commands Reference Manual*

The *UNICOS User Commands Ready Reference*, publication SQ–2056, accompanies the *UNICOS User Commands Reference Manual*.

The *UNICOS Administrator Commands Ready Reference*, publication SQ–2413, accompanies the *UNICOS Administrator Commands Reference Manual*.

The following manuals contain collections of man pages that describe the UNICOS library routines:

Publication	Title
SR–2079	*UNICOS Fortran Library Reference Manual*
SR–2080	*UNICOS C Library Reference Manual*
SR–2081	*Scientific Libraries Reference Manual*
SR–2138	*Math Library Reference Manual*

In some cases, man pages associated with a given product are published in the documentation set for that product, rather than in the UNICOS manuals listed here. For more information about the availability and content of any Cray Research publication, see the *User Publications Catalog*, publication CP–0099.

Ordering publications

The *User Publications Catalog*, publication CP–0099, lists all Cray Research hardware and software manuals that are available to customers.

To order a manual, either call the Distribution Center in Mendota Heights, Minnesota, at (612) 683–5907 or send a facsimile of your request to fax number (612) 452–0141. Cray Research employees may choose to send electronic mail to `order.desk` (UNIX system users) or `order desk` (HPDesk users).

Reader comments

If you have comments about the technical accuracy, content, or organization of this manual, please tell us. You can contact us in any of the following ways:

- Send us electronic mail from a UNICOS or UNIX system, using the following UUCP address:

  `uunet!cray!publications`

- Send us electronic mail from any system connected to Internet, using the following Internet addresses:

  `pubs2507@timbuk.cray.com` (comments specific to this manual)

  `publications@timbuk.cray.com` (general comments)

- Contact your Cray Research representative and ask that a Software Problem Report (SPR) be filed. Use `PUBLICATIONS` for the group name, `PUBS` for the command, and `NO-LICENSE` for the release name.

- Call our Software Information Services department in Eagan, Minnesota, through the Technical Support Center, using either of the following numbers:

  (800) 950–2729 (toll free from the United States and Canada)

  (612) 683–5600

- Send a facsimile of your comments to the attention of "Software Information Services" in Eagan, Minnesota, at fax number (612) 683–5599.
- Use the postage-paid Reader's Comment form at the back of this manual.

We value your comments and will respond to them promptly.

Contents

	Page
Preface	iii
Related publications	iii
Conventions	iv
Man page references	v
Ordering publications	vii
Reader comments	vii
CRAY T3D Administrative Concepts [1]	1
Administrative resource pools	1
Group ID pools	2
BATCH, INTERACTIVE, or BOTH pools	2
AVAILABLE or UNAVAILABLE pools	3
Express job pools	3
Network routing tables	4
Creating network routing tables	4
Reconfiguring network routing tables	5
MPP daemon (mppd(8))	5
CRAY T3D man pages	6
CRAY T3D Configuration Planning [2]	9
Updating the UNICOS parameter file	10
Modifying the UNICOS kernel tables	11
Choosing a shape for an administrative resource pool	11
Determining the space needed for system dumps	12
Setting limits for express processing	12
Maintaining a CRAY T3D System [3]	15
Changing attributes of administrative resource pools	15
Draining administrative resource pools	16
Shutting down the CRAY T3D system	17
Rebooting the CRAY T3D system	18
Monitoring a CRAY T3D System [4]	21
Monitoring CRAY T3D system activity	21
Monitoring active CRAY T3D applications	22
Monitoring PE status	23
Monitoring CRAY T3D resources	24
Monitoring NQS status	25
Troubleshooting a CRAY T3D System [5]	27
CRAY T3D troubleshooting strategy	27
Examining CRAY T3D system log files	28
Performing a dump of CRAY T3D system memory	29
CRAY T3D System Messages [A]	31
CRAY T3D Man Pages [B]	39
mppexec(1)	41
blt_copy(2)	44
mppconfig(5)	47
mppsyslog(5)	51
mppboot(8)	54

	Page
mppcmd(8)	56
mppd(8)	58
mppping(8)	62
mpproute(8)	65
mppstart(8)	68
mppstat(8)	70
mppsysdmp(8)	73
olnx(8)	76
olperi(8)	78

CRAY T3D Administrative Concepts [1]

CRAY T3D Administrative Concepts [1]

This section describes concepts necessary for understanding how to administer a CRAY T3D system. These concepts include the following:

- Administrative resource pools
- Network routing tables
- MPP daemon (`mppd`(8))

This section also references the UNICOS man pages that have been modified to accommodate CRAY T3D systems and new man pages of interest to administrators of CRAY T3D systems. Copies of the CRAY T3D man pages are included in appendix B.

Administrative resource pools

1.1

The UNICOS MAX operating system provides a multiuser environment through the use of *space sharing*, which allows multiple applications to run concurrently in separate partitions on the CRAY T3D system. To provide space sharing, a system administrator divides the CRAY T3D processing elements (PEs) into *administrative resource pools*. A user requests a partition from within a given administrative resource pool.

A system administator can control the type of processing that is allowed on the CRAY T3D system by dividing the PEs into administrative resource pools. These pools are set up during the system configuration process, at CRAY T3D boot time.

Each pool has a set of attributes that the system administrator can use to restrict the type of application that may use the PEs in that pool, to control whether or not an application may initiate using the PEs in that pool, and to affect job queuing and scheduling. For a complete list of possible attributes for a CRAY T3D administrative resource pool, see the `mppconfig`(5) man page.

The administrative resource pool attributes that restrict the type of application that may use the PEs in a pool are as follows:

- Group identification numbers (GIDs)
- BATCH
- INTERACTIVE
- BOTH (batch and interactive)

The administrative resource pool attributes that control whether or not an application may initate, using the PEs in that pool, are as follows:

- AVAILABLE
- UNAVAILABLE

The administrative resource pool attributes that can be used to affect job queuing and scheduling are as follows:

- ExpressTime
- MaxWaitTime

The following subsections describe these pool attributes.

Group ID pools 1.1.1

An administrative resource pool can be marked with one or more group ID numbers, limiting access to the pool to users who are members of the specified groups. The absence of a group ID on a pool indicates that all groups may use the pool.

The default attribute is no group ID, meaning that all groups may use the administrative resource pool.

BATCH, INTERACTIVE, *or* BOTH *pools* 1.1.2

An administrative resource pool can be designated as BATCH (batch only), INTERACTIVE (interactive only), or BOTH (batch and interactive).

Designating the pool as BATCH allows only applications submitted through the Network Queuing System (NQS) to use the resources of the pool. A batch-only administrative resource pool allows the scheduling and queuing mechanisms of NQS to be applied to the MPP batch resources.

Designating the pool as INTERACTIVE allows only applications inititated by users interactively connected to the Cray Research host system to use the resources of the pool. An interactive-only administrative resource pool is typically used when developing or debugging an application.

Designating the pool as BOTH allows the resources of the pool to be used both by applications submitted through NQS and by applications initiated by users interactively connected to the Cray Research host system.

The default attribute is BOTH (batch and interactive).

AVAILABLE *or* UNAVAILABLE *pools* 1.1.3

An administrative resource pool can be designated as either AVAILABLE or UNAVAILABLE.

The configuration driver allocates a partition only from an administrative resource pool that has the AVAILABLE attribute. No application initiates until it can be assigned to an available administrative resource pool.

Designating an administrative resource pool as UNAVAILABLE signals the configuration driver not to allocate any new resources from this pool. This does not affect applications that are already running using resources from this pool.

The default attribute is AVAILABLE.

Express job pools 1.1.4

By default, an administrative resource pool processes jobs on a first-in, first-out (FIFO) basis. However, the system administrator can set up the pool to allow small jobs to initiate ahead of large jobs. This *express* processing is enabled by setting the ExpressTime and MaxWaitTime attributes to nonzero.

The ExpressTime attribute sets an upper limit on the time (in seconds) that a job can run and still be considered a small job, and therefore a candidate for processing ahead of larger jobs. The MaxWaitTime attribute ensures the scheduling of jobs larger than this by setting an upper limit on the time (in seconds) that a large job can be starved for resources while small jobs are moved ahead.

The default value for the ExpressTime and MaxWaitTime attributes is 0 (no effect).

Network routing tables
1.2

The topology of the MPP interconnect network has each node of the system connected in a 3-D torus network, such that each node connects to neighboring nodes in the X, Y, and Z dimensions (both positive and negative directions). This allows each compute node to communicate directly to six neighboring nodes. The compute nodes are connected in three dimensions (X, Y, and Z), and the I/O gateways are connected in two dimensions (X and Y).

Traversal of the CRAY T3D interconnect network is done through *dimension-order routing*. All network traffic travels first in the X dimension (either positive or negative), then turns into the Y dimension (either positive or negative), and finally turns into the Z dimension.

The interconnect network steers memory request and response packets between processing elements (PEs) in a partition by using relative addressing based on the PE number. The virtual-to-physical conversion process must translate a virtual PE number into a physical PE number and represent the value as relative directions and distances to travel in the network.

The resulting relative PE address is the *routing tag* of the network packet. The routing tag contains the same information as the physical PE number, but it is organized into delta-x, delta-y, and delta-z fields. The PE number-to-routing-tag conversion affects only references to remote memory.

Each PE in the CRAY T3D system has a unique *routing table* that contains routing tags for all other PEs in the network. The routing tables are different for each PE because the delta-x, delta-y, and delta-z values are relative to the logical location in the torus. To find an entry for a particular PE in the routing table, the hardware takes the logical PE number and uses it as an index into the table to pull out the routing tag stored at that position.

Creating network routing tables
1.2.1

A CRAY T3D system administrator creates network routing tables on the Cray host system by using the `mpproute`(8) command. The `mpproute` command generates routing tables for each of the PEs using a CRAY T3D configuration file previously generated by the system administrator. This file includes any compute node failures, downed links, and barrier circuit failures. A default configuration file is provided as part of the CRAY T3D software package.

Reconfiguring network routing tables
1.2.2

When a CRAY T3D component fails (either a node or a PE stops responding or a network switch fails), a reconfiguration of the network routing tables is triggered either by the CRAY T3D system signaling the Cray host system to take action or by a system administrator on the Cray host system requesting the reconfiguration directly. Reconfiguration requires rebooting the CRAY T3D system. After the new routing tables are created, the information held in them is passed to the CRAY T3D system when a system administrator uses the `mppstart`(8) command (when the CRAY T3D system is booted).

MPP daemon (`mppd`(8))
1.3

UNICOS MAX software includes the `mppd`(8) utility, a multitasked daemon process that performs the following functions:

- Handles any user request initiated by sending a request over the named pipe `/usr/spool/mpp/mppd.regpipe` and any internal request originating from the MPP daemon error logger task through a shared-memory mechanism.
- Monitors and logs all MPP system activity to the MPP system daemon log (`/usr/spool/mpp/mppd.log`).
- Ensures that all partitions are freed after the `mppexec`(1) process has exited.

For detailed information about the MPP daemon, see the `mppd`(8) man page.

CRAY T3D man pages
1.4

This subsection references both the UNICOS man pages that have been modified to accommodate CRAY T3D systems and CRAY T3D man pages of interest to administrators of CRAY T3D systems.

Many UNICOS man pages have been changed to accommodate CRAY T3D systems. For details of the changes, use the man(1) utility to review the man page.

UNICOS man pages that contain changes of particular interest to administrators of CRAY T3D systems include the following:

Man page	Change
ps(1)	Added the -m and -M options
qstat(1)	Added the -m and -M options
qsub(1)	Added the -l option
limit(2)	Added the following resource definitions: L_MPPB, L_MPPE, and L_MPPT
udblib(3)	Added fields for the following limits: ue_jpelimit, ue_jmpptime, ue_jmppbarrier, ue_pmpptime, jpelimit, jmpptime, jmppbarrier, and pmpptime
nu(8)	Added the following directives: DefaultPe, DefaultMt, DefaultMb, and DefaultPt
qmgr(8)	To the rt[ime_limit] subcommand, added the following values for *requestid*: mpp_blimit, mpp_plimit, mpp_tlimit, and p_mpp_tlimit. Added the following qmgr subcommands: se[t] g[lobal] mpp_b[arrier_limit] [=] *limit* se[t] g[lobal] mpp_p[e_limit] [=] *limit* se[t] per_p[rocess] mp[p_time_limit] = *limit queue* se[t] per_r[equest] mpp_b[arrier_limit] = *limit queue* se[t] per_r[equest] mpp_p[e_limit] = *limit queue* se[t] per_r[equest] mpp_t[ime_limit] = *limit queue*
udbgen(8)	Added fields for the following limits: jmppbarrier, jmpptime, jpelimit, and pmpptime

In addition, man pages have been created to document utilities, scripts, and file formats unique to CRAY T3D systems. To view these CRAY T3D man pages online, use the man(1) utility. To view a hard copy of these pages, see appendix B.

The following man pages document utilities, scripts, system calls, and file formats that are of special interest to administrators of CRAY T3D systems:

Man page	Description
mppexec(1)	Initiates and services a user application on a CRAY T3D system
blt_copy(2)	Performs a data transfer using the CRAY T3D system block transfer engine
mppconfig(5)	MPP configuration file format
mppsyslog(5)	MPP system log file
mppboot(8)	Configures and boots Cray MPP systems
mppcmd(8)	Sends a request to the MPP daemon
mppd(8)	Starts the MPP daemon
mppping(8)	Tests the MPP gateway connections and compute processing elements (PEs)
mpproute(8)	Generates MPP binary configuration file with routing tables
mppstart(8)	Initiates the MPP deadstart sequence
mppstat(8)	Displays MPP resource status
mppsysdmp(8)	Dumps CRAY T3D system memory
olnx(8)	Tests CRAY T3D interconnect network hardware
olperi(8)	Tests CRAY T3D processor chip user mode instructions

CRAY T3D Configuration Planning [2]

CRAY T3D Configuration Planning [2]

This section describes some considerations that should be taken into account either when planning your initial CRAY T3D system configuration or when planning a reconfiguration after the system is installed and running. These include both UNICOS considerations on the Cray Research host system and UNICOS MAX considerations on the CRAY T3D system.

- Updating the UNICOS parameter file
- Modifying the UNICOS kernel tables
- Choosing a shape for an administrative resource pool
- Determining the space needed for system dumps
- Setting limits for express processing

The following subsections discuss each of these considerations.

Updating the UNICOS parameter file 2.1

The UNICOS parameter file (`/etc/config/param`), located on the Cray Research host system, contains entries that apply specifically to the CRAY T3D system. These entries are as follows:

- Low-speed (LOSP) channels to the I/O gateways.
- Number of I/O gateways (`GATEWAYS`).
- Number of buffer headers desired (`NTRANSACT`). Transmission Control Block (TCB) maps to the UNICOS buffer header. Packet Control Block (PCB) is the LOSP packet with control information attached.
- Number of YPE devices (= number of partitions) (`NYPEDEV`).
- Number of partitions (`NPARTITION`).
- Number of pools (`NPOOL`).
- Primary high-speed (HISP) buffer size (for all primary data) (`PBUFSIZE`).
- Secondary HISP buffer size (to transfer all data about the system call from the application to the agent) (`SBUFSIZE`).

The CRAY T3D entries in the UNICOS parameter file are updated during the CRAY T3D installation and configuration process. This occurs automatically, when configuration changes made using the menu system are activated.

In a sample UNICOS parameter file, the entries related to the CRAY T3D system appear as follows:

```
mainframe {
        channel 030 is lowspeed to gateway 0;
        channel 032 is lowspeed to gateway 1;
}
mpp {
        2   GATEWAYS;
        500 NTRANSACT;
        32  NYPEDEV;
        16  NPARTITION
        7   NPOOL;
        100 blocks PBUFSIZE;
        40  blocks SBUFSIZE;
}
```

Modifying the UNICOS kernel tables
2.2

The following UNICOS kernel tables may need to be increased in size:

- Process table size (NPROC). Default is 650.
- Maximum number of in-core file structures (NFILE). Default is 2100.
- Maximum number of in-core inodes (NINODE). Default is 1500.
- Maximum number of open files per process (OPEN_MAX). Default is 64.
- Maximum number of open files per process (OPEN_MAX). (< /usr/src/uts/include/sys/param.h). Default is 64.

Choosing a shape for an administrative resource pool
2.3

When choosing a shape for an administrative resource pool, you must choose the same shape that the configuration driver uses to search for a partition of the same size. For example, to configure a four-node pool, you must match the shape used by the configuration driver to search for four-node partitions.

For each cabinet type, the shape of each size partition is specified in a table. The tables are defined in the mpp_barrier.h file. For example, the table for an MCA128 cabinet type is as follows:

```
static shape_t BAR_SHAPE_MCA128 [] = {
1, 1, 1,   /* shape for partition of 1 node   (0)*/
1, 2, 1,   /* shape for partition of 2 nodes  (1)*/
2, 2, 1,   /* 1/2 shape for partition of 4 nodes */
2, 2, 2,   /* shape for partition of 8 nodes  (3)*/
4, 2, 2,   /* shape for partition of 16 nodes (4)*/
4, 2, 4,   /* shape for partition of 32 nodes (5)*/
4, 4, 4,   /* shape for partition of 64 nodes (6)*/
0, 0, 0,   /* (7) */
0, 0, 0,   /* (8) */
0, 0, 0,   /* (9) */
0, 0, 0,   /* (10) */
0, 0, 0,   /* (11) */
0, 0, 0,   /* (12) */
0, 0, 0,   /* (13) */
0, 0, 0,   /* (14) */
0, 0, 0,   /* (15) */
}
```

For an MCA128 cabinet, the shape for a one-node partition is 1-by-1-by-1, the shape for a two-node partition is 1-by-2-by-1, and so on. To configure a four-node pool, you must choose the shape of a four-node partition (2-by-2-by-1). If you choose any other shape for the four-node pool, a four-node (8-PE) application will never be scheduled to run in the pool.

Determining the space needed for system dumps

2.4

The recommended amount of reserved space is 1.5 times the amount of processing element (PE) memory on the CRAY T3D system. When you create a dump of the CRAY T3D system memory using the `mppsysdmp`(8) utility, each PE yields 4 Mbytes of disk space. The UNICOS MAX agent core (if dumped) requires 1 Mbyte of disk space.

Setting limits for express processing

2.5

When scheduling for the CRAY T3D system, the configuration driver looks at waiting jobs on a first-in, first-out (FIFO) basis for each administrative resource pool. When a job is found that is eligible for the resources of a pool, but that cannot run because resources are not available (for example, if the job requests 128 PEs and only 64 are available), normal scheduling is stopped for that pool.

Once all the pools have been scheduled on a FIFO basis, the pools are checked again for jobs that qualify for express processing. The concept of express processing assumes that some number of small jobs can run without impacting the big jobs. The configuration driver then looks at waiting jobs that are eligible for express processing.

To be considered for express processing, a job must have an MPP process time limit lower than the `ExpressTime` limit for a given pool. The `ExpressTime` limit is specified in the CRAY T3D system configuration file (see `mppconfig`(5)). The job's MPP process time limit can be set by using the `mppexec -time` option, by using a `qsub` directive, or through the user database (UDB) limits for a user. The configuration driver uses the most restrictive (smallest) value in determining whether or not the job qualifies as an express job.

This value is then used and enforced as the application time limit. When the application exceed its time limit, the application is killed.

The `MaxWaitTime` attribute of an administrative pool ensures that the large jobs are not held in the queue indefinitely by the smaller jobs qualifying for express processing. The `MaxWaitTime` limit is specified in the CRAY T3D system configuration file (see the `mppconfig`(5) man page).

The value assigned to the `MaxWaitTime` attribute determines the maximum time (in seconds) that a large job can be starved out by express scheduled jobs. After the large job has waited in the pending queue for `MaxWaitTime` seconds, express processing is suspended for that pool until the large job has been initiated.

If `ExpressTime` and `MaxWaitTime` are both zero (the default), or if either is zero, neither takes effect and the configuration driver uses a priority queuing (FIFO) scheduling algorithm.

To let a small job be scheduled ahead of a waiting large job, set both the `ExpressTime` and the `MaxWaitTime` attributes to large numbers. For more information and an example, see the `mppconfig`(5) man page.

Maintaining a CRAY T3D System [3]

Maintaining a CRAY T3D System [3]

This section describes the tasks performed by the administrator of a CRAY T3D system to stop, restart, and otherwise maintain the system. These tasks include the following:

- Changing attributes of administrative resource pools
- Draining administrative resource pools
- Shutting down the CRAY T3D system
- Rebooting the CRAY T3D system

Changing attributes of administrative resource pools

3.1

Attributes assigned to an administrative resource pool are used to restrict the type of application that may use the processing elements (PEs) from that administrative resource pool. Attributes of administrative resource pools are defined in the CRAY T3D configuration file (`/mpp/cf/config.local`). For a complete list of possible attributes, see the `mppconfig`(5) man page.

During CRAY T3D system installation, the system administrator uses the UNICOS 8.0 Installation / Configuration Menu System (the menu system) to assign the desired attributes to the pools that have been created. When the CRAY T3D system is booted, the menu system updates the CRAY T3D configuration file.

Do not edit the CRAY T3D configuration file directly; always use the menu system to make permanent changes to the attributes of administrative resource pools.

Attributes of an administrative resource pool can be changed either temporarily (for the current boot cycle) or permanently.

To change temporarily the current attributes of a pool, use the `mppcmd`(8) interface to the CRAY T3D daemon, `mppd`(8). To remove attributes, use the `clear` option:

`mppcmd clear` *poolid attribute* [*attribute*]

To add attributes, use the `set` option:

```
mppcmd set poolid attribute [attribute]
```

When the CRAY T3D system is rebooted, changes that were implemented using the mppcmd utility will be ignored.

To change permanently the current attributes of a pool, use the menu system:

```
Configure system ==>
   MPP Configuration submenu ==>
      UNICOS MAX ==>
         Software Pool Attributes
```

Any changes made using the menu system will take effect when the CRAY T3D system is rebooted and the menu system updates the CRAY T3D configuration file. For more information about the menu system, see the *UNICOS MAX Installation Guide*, publication SG–5216.

Draining administrative resource pools

3.2

Draining an administrative resource pool is the method by which a system administrator changes that pool from an active processing state to a quieted state. Draining pools is necessary for gracefully shutting down the CRAY T3D system or for changing the layout or attributes of the pools.

The administrator drains a pool by changing the attributes of that pool from available to unavailable. After the pool is marked as unavailable, no new applications are allowed to initiate using the resources in that pool. All applications currently running in that pool are allowed to complete. The pool is then said to be drained.

To gracefully shut down the CRAY T3D system, the system administrator must use the mppcmd(8) utility to mark all pools as unavailable:

```
mppcmd set all UNAVAILABLE
```

To change the layout or attributes of administrative resource pools, the system administrator must mark only the affected pools as unavailable:

```
mppcmd set poolid UNAVAILABLE
```

Shutting down the CRAY T3D system
3.3

Shutting down the CRAY T3D system means terminating the processes related to the operation of the CRAY T3D system. The CRAY T3D system can be shut down without affecting operation of the Cray Research host system.

Typically, you might want to shut down the CRAY T3D system to perform routine maintenance work. Other reasons to do so include to reenable a downed processing element or to reboot the system using a different configuration file.

To enable a graceful shutdown of both the CRAY T3D system and the Cray Research host system, the system administrator will typically embed the CRAY T3D shutdown command in the UNICOS shutdown script (`/etc/shutdown`).

To shut down the CRAY T3D system gracefully, first drain all administrative resource pools:

```
mppcmd set all UNAVAILABLE
```

Then issue the following command:

```
mppcmd shutdown grace_period
```

For example, the following command causes the CRAY T3D system to be shut down following a grace period of 60 seconds:

```
mppcmd shutdown 60
```

This command sends a `SIGSHUTDN` signal to all running agents. After 60 seconds, all agents are terminated (using a `SIGKILL` signal) and the CRAY T3D system is master-cleared.

Rebooting the CRAY T3D system
3.4

To invoke most changes to the CRAY T3D system, the system must be rebooted. To reboot the CRAY T3D system, use the `mppboot`(8) command. The `mppboot` command does the following:

1. Executes the `mpproute`(8) command to generate network routing tables. The `mpproute` command looks for a file named `mppconfig.local`, which contains information about bad nodes and downed links. The `mpproute` command then generates a binary routing table named `mpp.route`. This file contains a unique routing table for each node in the CRAY T3D system. The routing table is used as input to the `mppstart`(8) command.

2. Deadstarts the CRAY T3D system using `mppstart`(8). The `mppstart` command reads the routing table for desired system characteristics. It then issues a master clear over the low-speed channel (LOSP) to the deadstart node for the CRAY T3D system and downloads, to the deadstart node, a copy of the primary boot Privileged Architecture Library (PAL), routing tables, I/O node control software, system PAL, and microkernel binary. The deadstart node then propagates these binaries to the appropriate nodes of the CRAY T3D system. The `mppstart` command can also be used to perform a partial reboot.

3. Starts the CRAY T3D daemon process (`mppd`(8)).

4. Initializes the CRAY Y-MP resource allocation driver (the configuration driver).

The following command performs a simple reboot using the default CRAY T3D configuration file (`mppconfig.local`) in the current working directory:

```
mppboot
```

The following command performs a CRAY T3D reboot using a specified configuration file. The result is that a new routing table is generated from this file.

```
mppboot -c /mpp/cf/typhoon
```

The following command performs a reboot using a specified deadstart device. If you do not specify a deadstart device, one is chosen for you from among the I/O gateway devices.

```
mppboot -g /dev/mpp/iog02
```

Either of the following commands performs a partial reboot of the CRAY T3D system. This means that only `admpal` (the primary boot PAL binary) is booted on the PE nodes. This action preserves memory contents on the PEs and allows the system to be dumped.

```
mppboot -p
mppstart -p
```

The following command boots the CRAY T3D system using a specified routing file. However, in contrast to the preceding example, a new route file is not generated, even if the CRAY T3D configuration file has been modified.

```
mppboot -R /mpp/cf/mpp.route
```

For more information about the `mppboot` command, including sample output, see the `mppboot`(8) man page.

Monitoring a CRAY T3D System [4]

Monitoring a CRAY T3D System [4]

This section describes the work of checking the general activity level and health of the CRAY T3D system. Monitoring the CRAY T3D system includes the following tasks:

- Monitoring CRAY T3D system activity
- Monitoring active CRAY T3D applications
- Monitoring PE status
- Monitoring CRAY T3D resources
- Monitoring NQS status

Monitoring CRAY T3D system activity

4.1

To monitor general CRAY T3D system activity, use the UNICOS `tail`(1) command to display the CRAY T3D log files (`mppsyslog` and `mppd.log`).

To monitor the system log file (`mppsyslog`), enter the following:

```
tail -f /usr/spool/mpp/mppsyslog
```

To monitor the MPP daemon log file (`mppd.log`), enter the following:

```
tail -f /usr/spool/mpp/mppd.log
```

For more information about analyzing the CRAY T3D log files, see subsection 5.2, "Examining CRAY T3D system log files," page 28, and the `mppsyslog`(5) man page.

Monitoring active CRAY T3D applications
4.2

One indicator of the activity level of the CRAY T3D system is the status of currently active CRAY T3D applications.

To display the status of all active CRAY T3D processes, and the partition with which each is associated, use the following UNICOS command:

```
ps -elm
```

To display information about CRAY T3D partitions only, use the following UNICOS command:

```
ps -elM
```

When you specify the `ps` command with the `-m` or `-M` option, you receive a process status report of all active UNICOS processes, with the following fields added:

Field	Description
`ETIME`	Wall-clock execution time for the CRAY T3D process.
`PEs`	Number of processing elements (PEs) allocated to the CRAY T3D process.
`PRTN`	CRAY T3D partition identification number of the process.
`SHAPE`	CRAY T3D partition shape (X:Y:Z); hardware partition only. This field prints only when the `-w` or `-l` option is specified, causing wide or long listing mode.
`STATE`	CRAY T3D partition state (`ACTIVE`, `ERROR`, `FROZEN`, `UNKNOWN`, `WAIT`, or `ZOMBIE`).
`TYPE`	CRAY T3D partition type (`HW`=hardware or `OS`=operating system).

Monitoring PE status
4.3

To determine the status of the CRAY T3D processing elements (PEs), use the `mppping`(8) command. The `mppping` command first polls for active I/O gateways, attempting to send an echo packet to both the input and output sides of each of the specified gateways, to see if the gateway responds. If no gateways are specified, `mppping` sends echo packets to all enabled gateways. If at least one I/O gateway responds, the `mppping` command sends a request to each configured compute PE. The response to this request is then used to determine whether the microkernel on that PE is up or down.

To determine the status (up or down) of only the compute PEs, use the following command:

```
mppping -p
```

To learn more about the configured compute PEs, use the following command:

```
mppping -v
```

This verbose mode of the `mppping` command displays a list that includes an entry for each configured compute PE, indicating whether the microkernel on that PE is up or down, and the state of the PE. Valid PE states are as follows:

PE state	Description
`idle`	PE is idle
`halted`	PE is halted
`booting`	PE is being booted
`user init`	User is being downloaded
`user startup`	User thread being started
`user running`	User is running
`user exit`	User is exiting

For more information about determining the status of PEs, and for examples of the use of the `mppping` command, see the `mppping`(8) man page.

Monitoring CRAY T3D resources
4.4

To review the current allocation of CRAY T3D resources, use the `mppstat`(8) command. The `mppstat` command can be used to display information about administrative resource pool usage, active user partitions, and configuration driver statistics.

Administrative pool usage information displayed includes configuration information (such as torus dimensions, redundant nodes, maximum pools, and pools in use) and specific information for each pool in use (such as attributes, flags, GIDs, member count, partitions from the pool, pool shape, and total and available nodes in the pool).

Active user partition information displayed includes state, type, owner, group, owning process, source pool, elapsed time, application name, logical partition node shape, and nodes in the partition.

Configuration driver statistics displayed include successful allocations, failed allocations, active requests, and pending requests.

To display information about both pool usage and active user partitions, enter the following:

```
mppstat -a
```

To display information only about pool usage, enter the following:

```
mppstat -P
```

To display information only about active user partitions, enter the following:

```
mppstat -p
```

For more information about monitoring CRAY T3D resources, and for examples of the use of the `mppstat` command, see the `mppstat`(8) man page.

Monitoring NQS status
4.5

One indicator of the health of the CRAY T3D system is the status of Network Queuing System (NQS) MPP queue and queue complex limits. To monitor this information, use the `qstat(1)` utility.

To display the MPP limits currently defined for each batch queue and the amount of that resource currently being used, enter the following command:

```
qstat -m
```

To display similar information for a queue complex, use the following command:

```
qstat -M
```

In these displays, each column has two entries separated by the character `/`. The first entry is the limit set for the queue or queue complex. The second entry is the current use. The characters `--` mean that no limit was set explicitly. The characters `**` mean that the maximum was specified.

The columns in the displays have the following meanings:

Column	Description
`QUEUE NAME`	Name of the queue
`QUEUE COMPLEX`	Name of the queue complex
`RUN`	Number of requests allowed to execute simultaneously in the queue or queue complex, followed by the number currently executing
`PE'S`	Maximum number of CRAY T3D processing elements (PEs) that all requests in the queue or queue complex are allowed to use at once, followed by the number currently in use
`BARRIERS`	Not implemented. Maximum number of CRAY T3D barriers that all requests in the queue or queue complex are allowed to use at once, followed by the number currently in use

For more information about monitoring NQS queues, see the `qstat`(1) man page.

Troubleshooting a CRAY T3D System [5]

Troubleshooting a CRAY T3D System [5]

This section describes activities performed by the CRAY T3D system administrator in order to isolate problems encountered in the day-to-day operation of the CRAY T3D system. This section includes the following topics:

- CRAY T3D troubleshooting strategy
- Examining CRAY T3D system log files
- Performing a dump of CRAY T3D system memory

CRAY T3D troubleshooting strategy

5.1

The CRAY T3D system is designed to meet a new strategy for troubleshooting Cray Research systems. In this strategy, the system administrator or analyst emphasizes problem isolation, rather than problem resolution. For the CRAY T3D system, the role of the system administrator or analyst changes from providing significant local analysis to observing, documenting, collecting evidence, and generating test cases. This methodology has been in place for some time for problems relating to compilers and to the I/O subsystem model E (IOS-E).

The CRAY T3D system troubleshooting strategy emphasizes problem isolation in part because the CRAY T3D system software is released only in binary form. As changes to source code are needed to repair problems, such changes will be performed by CRAY T3D software developers at Cray Research, in Eagan, Minnesota, rather than by system administrators or analysts in the field.

The CRAY T3D system troubleshooting strategy also emphasizes problem isolation because of the complexity of the system. CRAY T3D system administrators and analysts will develop expertise at distinguishing between CRAY T3D application problems and CRAY T3D system problems. For massively parallel processing systems in general, such problems are difficult to distinguish from one another.

For CRAY T3D system problems identified, CRAY T3D system administrators and analysts will develop expertise in distinguishing between CRAY T3D software problems and CRAY T3D hardware problems such as barrier network failures, PE node failures, and deadstart node failures.

CRAY T3D system problems experienced at sites should be reported using a Software Problem Report (SPR).

As experience becomes available using CRAY T3D systems with real-world MPP applications and resolving system problems encountered, step-by-step problem analysis tools can be created. At present, problem isolation relies on the following:

- Analyzing CRAY T3D system error messages (see appendix A)
- Examining CRAY T3D system log files
- Performing a CRAY T3D system dump

Examining CRAY T3D system log files

5.2

The CRAY T3D system includes three log files that can be helpful in determining what recent CRAY T3D system activity might have contributed to a problem. The log files are all kept in the `/usr/spool/mpp` directory.

Log file	Description
`mppsyslog`	The system log file stores messages created when the CRAY T3D system is booted, when partitions are allocated and freed, and when channel errors occur.
`mppsyslog.bin`	The binary message file holds complete dumps of error messages received by the `mppd`(8) error task.
`mppd.log`	The daemon log traces starting and stopping of the daemon and actions taken.

For examples of all three log files, see the `mppsyslog`(5) man page.

Performing a dump of CRAY T3D system memory
5.3

To create a dump of CRAY T3D system memory, use the `mppsysdmp`(8) utility. The `mppsysdmp` utility captures areas of processing element (PE) control software memory.

The `mppsysdmp` utility initiates a partial system boot and dumps the memory. The memory data is dumped to a set of files (one binary file per PE in the system) in a dump directory within the specified directory (`/core` by default).

After the `mppsysdmp` utility completes, reboot the CRAY T3D system normally, using the `mppstart`(8) utility.

For more information about the `mppsysdmp` utility and for an example of output of the `mppsysdmp` utility, see the `mppsysdmp`(8) man page.

CRAY T3D System Messages [A]

CRAY T3D System Messages [A]

This appendix documents messages issued by CRAY T3D system software, including:

- Configuration driver
- I/O gateway (IOG) driver
- Microkernel
- `mppd`
- `mppexec`
- `mppstart`

Each message is listed, along with an extended explanation. For each message, the severity level is indicated: Information, Error, or Warning. Action needed, in response to the message, is also provided. The messages are arranged in alphabetical order, first by the system software issuing the message, and then by the message.

The CRAY T3D system messages are **not** available through the `explain`(1) utility.

MPP – Configuration driver

`All pools marked.`
pool attribute `[set/cleared].`

Attributes on the administrative pools shown have been changed.
Severity: Information.
Action: None.

MPP – Configuration driver

`Appl` *applname* `(`*partid partid*`) has exceeded the process time limit.`
`Uid` *uid* `Pid` *pid* `Session id` *sid*

The wall-clock time accumulated for this partition (from the time the partition was allocated until now) exceeds the limit value stored in the process table for the `mppexec` process associated with this partition.
Severity: Information.
Action: None.

MPP – Configuration driver

`Appl` *applname* `(`*partid partid*`) has exceeded the session time limit.`
`Uid` *uid* `Pid` *pid* `Session id` *sid*

The wall-clock time accumulated for this partition (from the time the partition was allocated until now) exceeds the limit value stored in the session table for the `mppexec` process associated with this partition.
Severity: Information.
Action: None.

MPP – Configuration driver

`Appl` *applname* (*partid partid*) `has reached the process time limit.`
`Uid` *uid* `Pid` *pid* `Session id` *sid*

The wall-clock time accumulated for this partition (from the time the partition was allocated until now) is within 5 seconds of the limit value stored in the process table for the `mppexec` process associated with this partition.
Severity: Information.
Action: None.

MPP – Configuration driver

`Appl` *applname* (*partid partid*) `has reached the session time limit.`
`Uid` *uid* `Pid` *pid* `Session id` *sid*

The wall-clock time accumulated for this partition (from the time the partition was allocated until now) is within 5 seconds of the limit value stored in the session table for the `mppexec` process associated with this partition.
Severity: Information.
Action: None.

MPP – Configuration driver

`Configuration driver statistics:`
`Successful allocations:` *num1*
`Failed allocations:` *num2*

The configuration driver has allocated *num1* partitions since the previous CRAY T3D system reboot. It has failed to allocate resources for *num2* resource requests since the previous CRAY T3D system reboot.
Severity: Information.
Action: None.

MPP – Configuration driver

`Marking barrier wire` *circuit #* `bad throughout wiremat.`

A barrier wire has been marked bad throughout the barrier wiremat because the state of a bypass point cannot be determined.
Severity: Error.
Action: Contact the site administrator. The CRAY T3D system can continue to run but the global barrier resources available have been diminished until the CRAY T3D system is rebooted.

MPP – Configuration driver

`MPP Administrative Pools have been loaded.`

A new administrative resource pool configuration has been loaded into the configuration driver.
Severity: Information.
Action: None.

MPP – Configuration driver

`MPP barrier information has been loaded.`

The barrier information for the attached CRAY T3D system has been loaded into the configuration driver.
Severity: Information.
Action: None.

MPP – Configuration driver

`MPP configuration set to` *torus dim*`, with` *red* `redundant nodes.`

The attached CRAY T3D system configuration has been set in the configuration driver with the torus dimensions *xdim*:*ydim*:*zdim* and *red* redundant nodes in the chassis.
Severity: Information.
Action: None.

MPP – Configuration driver

`MPP redundant node mapping table has been loaded.`

The redundant node mapping table has been loaded into the configuration driver.
Severity: Information.
Action: None.

MPP – Configuration driver

`Operator accessing [Allow,Deny] list for pool` *poolid uid*,

Some action is being taken on the user ID list for the indicated pool. A *poolid* = –1 indicates all pools.
Severity: Information.
Action: None.

MPP – Configuration driver

`Operator accessing gid list for pool` *poolid gid*,

Some action is being taken on the group ID list for the indicated pool. A *poolid* = –1 indicates all pools.
Severity: Information.
Action: None.

MPP – Configuration driver

`Partition` *partid* `is released.`

A partition has been released. The nodes and barrier resources allocated to this partition have been marked "free."
Severity: Information.
Action: None.

MPP – Configuration driver

`Partition` *partid* `is sleeping for resources.`

The configuration driver has put a process to sleep waiting for resources to become available.
Severity: Information.
Action: None.

MPP – Configuration driver

`Partition` *partid* `has been allocated.`
`Application name :` *applname*
`Uid :` *username* (*uid*)
`Owning process id :` *pid*
`Partition type :` *partition type*
`Barrier circuit` *circuit* `# allocated, mask =` *mask value*
`Barrier bypass: PE` *PE#* `snc` *bit*
`Node count:` *node count*
`Nodes in partition :`
node list...

A partition has been allocated. The nodes and barrier resources allocated to this partition have been marked "in use."
Severity: Information.
Action: None.

MPP – Configuration driver

`Partition` *partid* `released (no resources held), error` *errno*

A process sleeping in the configuration driver for resources has been killed by the user or administrator.
Severity: Information.
Action: None.

MPP – Configuration driver

`Pool` *poolid* `marked.`
pool attribute `[set/cleared]`

Attributes on the administrative pools shown have been changed.
Severity: Information.
Action: None.

MPP – IOG driver

`Channel/Gateway` *channel ordinal* `had been disabled because of a protocol problem.`

`Logical Channel` *lchan*`, Sequence number` *seqn*`,`

The channel/gateway associated with the IOG device that has this minor device number has been disabled because of an internal protocol error. The error occurred on logical channel *lchan*.
Severity: Error.
Action: Contact site administrator. A CRAY Y-MP system dump or a CRAY T3D system dump may be required to resolve the problem. The CRAY T3D system may need to be rebooted to recover.

MPP – IOG driver

`Channel/Gateway` *channel ordinal* `has been disabled.`

The channel/gateway associated with the IOG device that has this minor device number has been disabled.
Severity: Error.
Action: Contact site administrator. A CRAY Y-MP system dump or a CRAY T3D system dump may be required to resolve the problem. The CRAY T3D system may need to be rebooted to recover.

MPP – IOG driver

`Channel/Gateway` *channel ordinal* `had been disabled because of a timeout.`

The channel/gateway associated with the IOG device that has this minor device number has been disabled because of a LOSP channel timeout.
Severity: Error.
Action: Contact site administrator. A CRAY Y-MP system dump or a CRAY T3D system dump may be required to resolve the problem. The CRAY T3D system may need to be rebooted to recover.

MPP – IOG driver

`Channel/Gateway` *channel ordinal* `has been enabled.`

The channel/gateway associated with the IOG device that has this minor device number has been disabled.
Severity: Information.
Action: None.

MPP – IOG driver

`The MPP has been master-cleared.`

The `IOG_MCLEAR` request has been completed.
Severity: Information.
Action: None.

MPP – IOG driver

`Channel/Gateway` *channel ordinal* `has received an input error on logical channel` *lchan*`.`

The channel/gateway associated with the IOG device that has this minor device number has received an input error on the logical channel indicated.
Severity: Warning.
Action: None.

MPP – IOG driver

`Channel/Gateway` *channel ordinal* `has received a retransmission request.`

The channel/gateway associated with the IOG device that has this minor device number has received a retransmission request from the CRAY T3D IOG.
Severity: Warning.
Action: None.

MPP – Microkernel

`LPE` *logpe* `: Hardware memory error signature` *signature*
`first` *cycle count* `last` *cycle count*
`count` *count*

The hardware on logical PE *logpe* experienced a memory error with the signature or syndrome *signature*. The message will indicate one or more single-bit ECC errors and was probably corrected by the hardware.
Severity: Warning.
Action: Contact the site administrator or site engineer.

MPP – Microkernel

`(LPE` *logical PE*`) : MPP_HARDWARE MISROUTE INDICATION,`

The logical PE indicated received a CRAY T3D system network packet that was misrouted.
Severity: Error.
Action: Contact the site administrator. The CRAY T3D system must be dumped and the dumps should be sent to Cray Eagan for further investigation.

MPP – Microkernel

`PANIC PPE` *phype* `(LPE` *logpe*`) :` *panic string*

The microkernel running on physical PE *phype* panicked with the printed panic string.
Severity: Error.
Action: Contact the site administrator. The CRAY T3D system must be dumped. The dumps and copies of the `mppsyslog` file should be sent to Cray Eagan for further investigation.

MPP – Microkernel

`PPE` *physical pe* `(LPE` *logical pe*`) :` *string*

The microkernel has noticed something unusual happening in the CRAY T3D system.
Severity: Error.
Action: Contact the site administrator. The CRAY T3D system must be dumped and the dumps should be sent to Cray Eagan for further investigation.

MPP – `mppd`

`mppd: Boot directory is` *boot dir*

This is the current working directory when the `mppstart` command is issued.
Severity: Information.
Action: None.

MPP – `mppd`

`mppd: Can't connect to dgdaemon.`

The MPP daemon (`mppd`) tried and failed to connect to the diagnostics daemon (`dgdaemon`). This message will appear whenever `mppd` is started and cannot connect.
Severity: Information.
Action: None.

MPP – mppd

`mppd: Connection lost to dgdaemon.`

The MPP daemon (mppd) lost its connection to the diagnostics daemon (dgdaemon).
Severity: Information.
Action: None.

MPP – mppd

`mppd: Connection made to dgdaemon.`

The MPP daemon (mppd) tried and succeeded in connecting to the diagnostics daemon (dgdaemon).
Severity: Information.
Action: None.

MPP – mppd

`Partition` *partid* `Agent core file copied to` *core file path*

An mppexec process has panicked and the mppd process has copied the core file into the UNICOS MAX core file directory.
Severity: Error.
Action: Contact the site administrator. The core file should be sent to Cray Eagan for further investigation.

MPP – mppd

`Partition` *partid* `Exit complete message received`

The microkernels have completed processing the force exit request and the resources of that partition may be released.
Severity: Information.
Action: None.

MPP – mppd

`Partition` *partid* `Force exit message sent to PE` *pe*

The mppd process is processing a ZOMBIE partition associated with a mppexec process that terminated abnormally. The mppd process is sending a message to virtual PE 0 of the partition indicating that the microkernels should cleanup this application.
Severity: Information.
Action: None.

MPP – mppd

`mppd: sanity: All PEs are responding.`

All PEs have checked in with the deadstart node after the CRAY T3D system boot.
Severity: Information.
Action: None.

MPP – mppd

`mppd: sanity: Deadman time out received on PEs:`
pe list
`mppd: sanity: Disabled nodes (`*node count*`) :`
node list

Severity: Warning.
Action: Contact the site administrator. The CRAY T3D system must be dumped. The dumps and copies of the mppsyslog file should be sent to Cray Eagan for further investigation.

MPP – mppexec

`Partition` *partid* `Agent Notice: Agent running`

The microkernels and mppexec process are all initialized and the user application is now running in this partition.
Severity: Information.
Action: None.

MPP – mppexec

`Partition` *partid* `Agent Notice: exiting normally`

The agent process associated with this partition is now exiting normally.
Severity: Information.
Action: None.

MPP – mppexec

`Partition` *partid* `Agent Notice: exiting via user signal`

The agent process associated with this partition has received a catchable signal and is exiting because of it.
Severity: Information.
Action: None.

MPP – mppexec

`Partition` *partid* `Agent PANIC:` *panic string*

The mppexec process has experienced a fatal error. This is a failure in the UNICOS MAX system code.
Severity: Error.
Action: Contact site administrator.

MPP – mppstart

`mppstart: Booting admpal` *binary pathname*
`mppstart: Booting pool A_POOL with kernel` *binary pathname*
`mppstart: Booting pool A_POOL with PAL` *binary pathname*
`mppstart: Booting I/O nodes with` *binary pathname*

These messages show the location of the binaries used to boot the CRAY T3D system.
Severity: Information.
Action: None.

MPP – mppstart

`mppstart: MPP boot sequence started, uid` *uid*`, tty` *tty device*
`mppstart: Booting with route file` *route file path*

The mppstart command has been executed. These messages show the user ID, TTY device, and route file for the invocation.
Severity: Information.
Action: None.

MPP – mppstart

`mppstart: Deadstart node is` *node*`, SROM version` *version*

This shows the node name for the deadstart node used for this boot and lists the SROM version number for this node.
Severity: Information.
Action: None.

MPP – mppstart

`mppstart: Device` *device pathname* `is I/O node` *node*

The CRAY T3D IOG node associated with the UNICOS device is received in the master clear response packet from the targeted CRAY T3D IOG.
Severity: Information.
Action: None.

MPP – mppstart

`mppstart: Issuing Master Clear on` *device pathname*

The boot command is issuing IOG_MCLEAR request on the indicated device.
Severity: Information.
Action: None.

MPP – `mppstart`

```
mppstart: Killing MPP applications
```

The active applications are about to be killed because of a CRAY T3D system reboot.
Severity: Information.
Action: None.

MPP – `mppstart`

```
mppstart: MPP boot sequence completed
```

The `mppstart` command has completed the command sequence necessary to boot the CRAY T3D system.
Severity: Information.
Action: None.

CRAY T3D Man Pages [B]

CRAY T3D Man Pages [B]

This appendix provides man pages for commands, scripts, system calls, and file formats of special interest to administrators of CRAY T3D systems. The following man pages are included:

- mppexec(1)
- blt_copy(2)
- mppconfig(5)
- mppsyslog(5)
- mppboot(8)
- mppcmd(8)
- mppd(8)
- mpping(8)
- mpproute(8)
- mppstart(8)
- mppstat(8)
- mppsysdmp(8)
- olnx(8)
- olperi(8)

These man pages can be viewed online by using the UNICOS man(1) command.

NAME

mppexec – Initiates and services a user application on a CRAY T3D system

SYNOPSIS

mppexec a.out [-base *node*] [-debug] [-nosleep] [-npes *n* | *n-m*] [-pool *pool_name*] [-shape *X:Y:Z*] [-time *seconds*] [-ypesim *host*:[*path*]] [*user_options*] [*user_args*]

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The processing done by mppexec can be hidden from the user by having the loader place a #!/mpp/bin/mppexec directive at the top of the a.out file.

Options that are restricted for use only by mppexec include the following:

[-base *node*] Causes the partition to be allocated such that virtual PE 0 will be located on the specified node. If the PEs on node are currently busy, the application will sleep waiting for them to be released. An example is as follows:

```
a.out -base 0x020 -npes 2
```

This example will assign the 2 PEs on node 0x020 to the application. If the PEs on node 0x020 are currently busy, the application will sleep waiting for them to be released.

-debug Specifies that the application should be downloaded but the start of execution should be postponed. This option is provided in support of the debugger and can be used only during an interactive session.

-nosleep Specifies that the user does not want to wait for resources to become available. By default, the user process in the kernel is put to sleep if the requested resources are currently in use.

-npes *n* or *n-m* Specifies the desired number of compute processing elements (PEs). PE resources can be specified either in terms of the total number of compute PEs required (*n*) or an allowable range of PEs (*n-m*).

If a range of PEs is specified, attempts will be made to allocate PEs starting with the largest value of the specified range down to the smallest value, until a partition is successfully allocated.

For the Fortran programming model, PEs are always allocated in powers of 2. Therefore, if the specified value does not result in a power of 2 PEs, the value is rounded up to the nearest power of 2 and a warning message is written to the user.

For the message-passing model, PEs are always allocated in pairs. Therefore, if an odd number of PEs is specified, the value is rounded up by 1 and a warning message is written to the user.

-pool *pool_name* Specifies the PE pool in which the application is to run. To obtain PE pool names and configuration information, use the mppstat(8) command.

`-shape` *X*`:`*Y*`:`*Z* — NOTE: Deferred implementation.
Specifies a shape hint in terms of the X:Y:Z dimensions. If the shape specified cannot be supported either by the hardware configuration or the programming model, the closest possible supported shape is selected and a warning message is written to the user.

`-time` *seconds* — Resets the process time limit for the application. The specified time must always be less than or equal to the UDB limits. Allows the user to control whether the job should be consided for express processing. The specified time will be used as the new CRAY T3D elapsed time limit for the process.

`-ypesim` *host*:[*path*] — Allows `mppexec` to be run under `mppsim`. For further information, see the *Cray MPP Simulator User's Guide*, publication SG–2503.

user_options — Options for the user's program.

user_args — Arguments for the user's program.

NOTES

When the `mppexec` process exits, the accouting record is complete. Although the resources might not be released immediately, the user is no longer charged.

ENVIRONMENT VARIABLES

Environment variables can be used in place of the command-line options. The environment variables supported include the following:

`MPP_NOSLEEP` — If nonzero, then do not sleep if resources are not available.

`MPP_NPES` — The number of PEs, specified either as the total number or as a range.

`MPP_POOL` — Specifies a specific PE pool in which the application is to run.

`MPP_SHAPE` — NOTE: Deferred implementation.
The shape hint for the partition specified in terms of X:Y:Z dimensions.

The UNICOS MAX agent services all I/O-related system calls issued by an MPP application. The following environment variables control the UNICOS MAX agent resource allocation:

`MPP_AGENT_IO_MEM_MIN`
Defines the initial amount of I/O memory to be allocated. Lower limit is 1 Mbyte. Upper limit is `MPP_AGENT_IO_MEM_MAX`, which causes all I/O memory space to be allocated at startup time. Default is 4 Mbytes.

`MPP_AGENT_IO_MEM_MAX`
Defines the upper limit of the UNICOS MAX agent's memory usage. Lower limit is `MPP_AGENT_IO_MEM_MIN`. Upper limit is `unlimited`, which means that the job or process memory limit applies. Default is `unlimited`.

`MPP_AGENT_IO_MEM_INC`
Provides a hint, when new memory resources are allocated. If the allocation fails, the UNICOS MAX agent reduces this value by half, until the allocation succeeds or the allocation size becomes smaller than the original request size. Lower limit is 1 Mbyte. Upper limit is `unlimited`, which means that the job or process memory limit applies. Default is 1.4 Mbytes.

`MPP_AGENT_IO_MEM_FREE_STRATEGY`
Defines the memory preallocation strategy. Possible values are as follows:

`never`	Allocated memory regions that become free will not be returned to UNICOS, but will be kept in the agent for later use.
`always`	Free memory regions are returned to the UNICOS kernel as soon as possible.
seconds	When a call to free up memory is allocated, waits the specified time before freeing up memory resources. Minimum is 1 second; maximum is 3600 seconds.

`MPP_AGENT_IO_CHUNK_SIZE`
The UNICOS MAX agent executes system calls on behalf of an MPP application. Data for a read system call, for example, is transferred from PE memory space into the UNICOS MAX agent, which then executes the read. The UNICOS MAX agent must provide buffer space for these operations. If a 32-PE application is reading 1 Mword simultaneously, the resulting memory requirement in the UNICOS MAX agent would be 32 Mwords. To reduce these requirements, I/O can be chunked into smaller pieces. `MPP_AGENT_IO_CHUNK_SIZE` defines this value. Requests smaller than the chunk size are handled in a single operation.

`MPP_AGENT_IOPATH`
Defines the I/O behavior. Possible values are as follows:

`buffered`	(Default) HISP I/O uses the kernel buffer for data transfers.
`directio`	HISP data can be moved directly from and to the UNICOS MAX agent address space.

`MPP_AGENT_PLOCK`
Controls physically locking the UNICOS MAX agent in memory. Possible values are as follows:

`none`	Does not physically lock the UNICOS MAX agent in memory.
`delay`	(Default) Physically locks the UNICOS MAX agent in memory, but does not move the agent into low memory at startup time.
`immediate`	Immediately moves the UNICOS MAX agent into low memory and physically locks the agent in memory at startup time.

`MPP_AGENT_SYSCALL_THREADS`
Defines the number of threads available to handle system call requests. The UNICOS MAX agent uses three threads for internal purposes. It also requires at least two system call threads, of which one is always waiting for new system call threads in order to queue them. The valid range is from 1 through 60 threads. Default is NPES/4.

SEE ALSO

`mppstat`(8)

Cray MPP Simulator User's Guide, publication SG–2503.

NAME

`blt_copy` – Performs a data transfer using the CRAY T3D block transfer engine

SYNOPSIS

```
long blt_copy (int mode, listbt_t *bl_list, int num_elements);
```

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The `blt_copy` system call uses the block transfer engine on the CRAY T3D system to perform the specified memory to memory copy. These copies are called block transfers, scatters, or gathers.

The arguments are as follows:

mode The action `blt_copy` performs:

- `BLT_START` Returns to the user as soon as the block transfer is initiated and does not wait for the block transfer to complete.
- `BLT_WAIT` Waits for the transfer to complete before returning to the user.

**bl_list* Pointer to `listbt_t`, the block-transfer structure type. `listbt_t` includes the following members:

```
typedef struct blt_transfer {
        long              *rmt_base;
        unsigned long     rmt_stride;
        unsigned long     rmt_index;
        unsigned long     rmt_mask;
        long              *lcl_base;
        unsigned long     lcl_stride;
        unsigned long     ivlength;
        long              type;
        long              *indx_vector;
        struct iosw       *lcl_sw;
        struct iosw       *rmt_sw;
        int               rmt_pe;
        } listbt_t;
```

- `rmt_base` The remote base address of the buffer.
- `rmt_stride` The stride in words at the remote processing element (PE).
- `rmt_index` The index into the buffer starting at the remote address.
- `rmt_mask` The mask used during the centrifuge operation to obtain the final remote offset and the PE number.
- `lcl_base` The local base address.
- `lcl_stride` The stride in words at the local PE.

`ivlength` The length of the transfer. The maximum value of `ivlength` under normal conditions is `MAXBLT`. Under special conditions, `ivlength` can exceed `MAXBLT` and be less than or equal to `MAXBLT*4` if and only if the following conditions are strictly adhered:

- `rmt_base` is cache-line aligned.
- `lcl_base` is cache-line aligned.
- The low-order two bits of the `rmt_mask` are zero.
- `rmt_stride` is equal to one.
- `lcl_stride` is equal to one.
- `ivlength` is a multiple of four.
- This is a constant stride transfer.

If the block transfer that is specified in the call to `blt_copy` adheres to all of the above rules, the transfer will actually use a cache-line transfer mode. The normal case is the single-word transfer mode.

`type` Transfer type (see `blt.h`).

`index_vector` The array of indices when the transfer type is `SCATTER` or `GATHER`. This address must be aligned on a cache line (32 byte) boundary.

`lcl_sw` The local status word that is updated when the block transfer is completed (see `iosw.h`).

`rmt_sw` A remote status word that can be updated when the block transfer is completed.

`rmt_pe` The PE where the remote status word address exists. This value must be set to the number of a valid PE if the application is running in an OS partition and the transfer type is `CSTRIDE_READ` or `CSTRIDE_WRITE`

num_elements The number of structures of type `listbt_t` that can be referenced through the `bl_list` pointer. This number must be less than `BLTMAXLIST`.

NOTES

The block transfer engine is a low-level hardware device and requires knowledge of the CRAY T3D hardware architecture. Since the block transfer engine is external to the local T3D processor, it does not have control over the processor's data cache. To maintain cache coherency on the local processor, `FLUSH_CACHE` for `bl_list->type` should be set so that the block transfer engine driver flushes the local cache upon transfer completion.

This setting does not help maintain cache coherency on a remote processor. Cache coherency on a remote processor can be maintained by locally flushing that processor's data cache.

If the sum total of all outstanding block transfers is greater than `BLTMAXLIST`, then the last `blt_copy` call will not return until the sum total of all outstanding calls is less than `BLTMAXLIST`.

`SCATTER` and `GATHER` are not valid transfer types in an OS partition.

RETURN VALUES

If `blt_copy` completes successfully, a value of 0 is returned; otherwise, a value of –1 is returned and `errno` is set to indicate the error.

The successful return of the call does not imply successful completion of the block transfer. The `blt_copy` system call returns a value in `errno` if it is unsuccessful in initiating a transfer. The block transfer engine driver returns an error condition in the status word(s) if it is unsuccessful in completing a transfer. Both the return value of `blt_copy` and the state of the status word(s) must be checked to guarantee that a transfer was initiated and completed successfully.

ERRORS

The `blt_copy` system call fails if one of the following error conditions occurs:

Error Code	Description
EINVAL	An argument that is not valid is passed to the system call.
E2BIG	*num_elements* is greater than BLTMAXLIST.
EFAULT	The address given for `lcl_base` or `rmt_base` in the `listbt_t` structure is not valid, or it is not a 64-bit word-aligned address: • The `indx_vector` address is not cache aligned. • The value specified for `rmt_pe` is not legal while the application is running in an OS partition. • A SCATTER or GATHER transfer type is specified while the application is running in an OS partition.
ERANGE	The `ivlength` value in the `listbt_t` structure is greater than MAXBLT.
ETIMEDOUT	The transfer failed due to a timeout. Any additional information may be obtainable through the status words.

If the block-transfer engine driver is unable to successfully complete a block transfer, it will update the `sw_flag` in the status word(s), indicate the `errno` in `sw_error`, and update the `sw_count` to indicate the number of bytes actually transferred. Possible return values that the application may see in the `sw_error` portion of the status word follow:

Error Code	Description
ENXIO	The block transfer attempted to access memory not accessible to the user.
ENODEV	The block transfer attempted to access memory on a PE that was not within the application's partition.
EIO	The block transfer suffered a memory error when attempting to read an indices from the `indx_vector` array.
ETIMEDOUT	The transfer failed due to a timeout. Any additional information may be obtainable through the status words.

SEE ALSO

CRAY T3D System Architecture Overview, publication HR–04033, for more information about the block transfer engine and transferring data

NAME

mppconfig – CRAY T3D system configuration file format

SYNOPSIS

/mpp/cf/config.local

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The default CRAY T3D system configuration file is config.local. This file is used as input to the mpproute(8) and mppstart(8) commands. This configuration file holds information regarding system configuration parameters, administrative resource pool layouts and attributes, node failures, and downed network links.

The following subsections present portions of a sample configuration file for a CRAY T3D system.

System Configuration

The system configuration portion of a sample CRAY T3D system configuration file is as follows:

```
Configuration {
   /*
    * system configuration
    */
   System {
      Serial  6001;
      Clock   150;
      Cabinet MC_256;
      Boot ADMPAL "/mpp/os/admpal";
      MaxPartition 16;
      IOM {
         0xC20;
         0xC02;
      }
   }
```

Parameter	Description
Serial	Serial number of the CRAY T3D system. NOTE: The Clock and Serial values are used by the target(2) system call.
Clock	Clock speed in megahertz.
Cabinet	Cabinet type defines the torus dimensions (X:Y:Z), the number of I/O modules (IOMs), and the number of redundant nodes.
Boot ADMPAL	Specify the admpal binary file with which to boot the system. The admpal binary is the primary boot Privileged Architecture Library (PAL) binary. The same admpal is used to boot each processing element (PE) in the system.
MaxPartition	Specify the maximum number of active partitions in the system at any one time.
IOM	If the configuration contains a nonstandard number of IOMs, list those included in this configuration.

I/O Gateway Configurations

The I/O gateway configurations portion of a sample configuration file is as follows:

```
Configuration {
   /*
    * I/O gateway configurations
    */
   Gateways {
      Select Closest;
      Boot kernel "/mpp/os/iog_os";
            gateway 0 {
           hisp mode c100d200;
            }
            gateway 1 {
           hisp mode c100d200;
            }
            gateway 2 {
           hisp mode c100d200;
            }
            gateway 3 {
           hisp mode c100d200;
            }
}
```

Parameter	Description
`Gateways`	The gateway numbers specified in the configuration file should match the low-speed (LOSP) channel definitions specified in the CRAY Y-MP UNICOS parameter file (`/etc/config/param`).
`Select`	Specify either `Closest`, `SamePlane`, or `RoundRobin`. Default is `RoundRobin`.
`Boot kernel`	Specify the I/O node control software binary file.

Administrative Resource Pools

The administrative resource pools portion of a sample configuration file is as follows:

```
Configuration {
   /*
    * compute PE pools
    */
   Pools {
      A_POOL {
         Boot kernel "/mpp/os/ukernel";
         Boot system pal "/mpp/os/maxpal";
         Attributes {
            GIDs mpp, os;
            Interactive;
            Batch;
            ExpressTime 30;
            MaxWaitTime 600;
            MaxPartition 8;
         }
         Compute Nodes {
            shape(0x000,8,4,4);
         }
      }
   }
}
```

Parameter	Description	
`Boot kernel`	Specify the microkernel binary file. If multiple pools are defined, specify the same microkernel binary file for each pool.	
`Boot system pal`	Specify the OS support PAL binary file. If multiple pools are defined, specify the same OS support PAL binary file for each pool.	
`Attributes`	Attributes include the following:	
	`GIDs`	List the group IDs of all groups allowed access to this pool. The default is all groups.
	Job types	Specify `BATCH`, `INTERACTIVE`, or `BOTH` (batch and interactive). The default is `BOTH` (batch and interactive).
	Availability	Specify `AVAILABLE` or `UNAVAILABLE`. The default is `AVAILABLE`.
	`ExpressTime`	Specify the maximum time (in seconds) that a job can run in order to be considered an express job (sets special scheduling considerations). The default is 0 (no effect).
	`MaxWaitTime`	Specify the maximum time (in seconds) that a nonexpress job would be starved for resources. Once any job has waited this time limit, express processing is suspended until this job is scheduled. The default is 0 (no effect).
	`MaxPartition`	Specify the maximum number of active partitions in this pool at any one time.

`Compute Nodes` Specify compute nodes either individually or in terms of shape (`shape (`*basenode*`,` *xwidth*`,` *ywidth*`,` *zwidth*`) ;`).

Hardware Failures

The hardware failures portion of a sample configuration file is as follows:

```
Hardware failures {
      BadNodes {
        0x102;        /* map a redundant node onto 0x102 */
      }
      BadLinks {
        0x108:x;      /* the x link between 0x108 and 0x10A is bad */
        0x108:y;      /* the y link between 0x108 and 0x10A is bad */
        0x230:z;      /* the z link between 0x230 and 0x232 is bad */
      }
      BadBarrier {
        circuit 0;    /* barrier circuit 0 is bad */
      }
}
```

Parameter	Description
`BadNodes`	All nodes listed will be logically replaced by redundant nodes. If a redundant node is specified as a bad node, it will not be used. If an I/O node is specified as a bad node, it will not be booted. If there are more bad nodes than redundant nodes, those nodes that cannot be replaced will be marked as `disabled` and will not be booted.
`BadLinks`	Bad network links are specified in terms of the node holding the positive side (direction) of the bad switch.
`BadBarrier`	Of the four (0–3) barrier circuits in the barrier wire mat, list all barrier circuits that are bad. If a barrier circuit has a failure point anywhere in the circuit, the whole circuit is disabled.

SEE ALSO

`mpproute`(8), `mppstart`(8)

`target`(2) in the *UNICOS System Calls Reference Manual*, publication SR–2012

NAME

mppsyslog – CRAY T3D system log file

SYNOPSIS

```
/usr/spool/mpp/mppsyslog
/usr/spool/mpp/mppsyslog.bin
/usr/spool/mpp/mppd.log
```

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The following log files are created and kept in the /usr/spool/mpp/ directory:

mppsyslog	CRAY T3D system log file. Stores messages created when the CRAY T3D system is booted, when partitions are allocated and freed, and when channel errors occur.
mppsyslog.bin	Binary message file (CRAY T3D log files). Holds complete dumps of error messages received by the mppd(8) error task.
mppd.log	CRAY T3D system daemon log file. Traces starting and stopping of the daemon and actions taken.

EXAMPLES

Example 1: A sample CRAY T3D system log file (mppsyslog) appears as follows:

```
02/04/94 16:45:28 - mppstart: MPP boot sequence started, uid 0, tty /dev/ttyp039
02/04/94 16:45:28 - mppstart: Booting with route file ./mpp.route
02/04/94 16:45:28 - mppstart: Killing MPP applications
02/04/94 16:45:28 - All pools marked.
02/04/94 16:45:28 -       MPP_UNAVAILABLE   set.
02/04/94 16:45:28 - Channel/Gateway 0 has been disabled.
02/04/94 16:45:28 - mppstart: Issuing Master Clear on /dev/mpp/iog00
02/04/94 16:45:28 - The MPP has been master-cleared.
02/04/94 16:45:28 - mppstart: Device /dev/mpp/iog00 is I/O node 0xc30
02/04/94 16:45:28 - Channel/Gateway 1 has been disabled.
02/04/94 16:45:28 - mppstart: Issuing Master Clear on /dev/mpp/iog01
02/04/94 16:45:28 - The MPP has been master-cleared.
02/04/94 16:45:28 - mppstart: Device /dev/mpp/iog01 is I/O node 0xc3e
02/04/94 16:45:28 - Channel/Gateway 2 has been disabled.
02/04/94 16:45:28 - mppstart: Issuing Master Clear on /dev/mpp/iog02
02/04/94 16:45:28 - The MPP has been master-cleared.
02/04/94 16:45:28 - mppstart: Device /dev/mpp/iog02 is I/O node 0xc18
02/04/94 16:45:28 - Channel/Gateway 3 has been disabled.
02/04/94 16:45:28 - mppstart: Issuing Master Clear on /dev/mpp/iog03
02/04/94 16:45:28 - The MPP has been master-cleared.
02/04/94 16:45:28 - mppstart: Device /dev/mpp/iog03 is I/O node 0xc16
02/04/94 16:45:28 - The MPP has been master-cleared.
02/04/94 16:45:28 - mppstart: Deadstart node is 0xc30, SROM version 2064
02/04/94 16:45:28 - mppstart: Booting admpal /mpp/os/admpal
02/04/94 16:45:34 - mppstart: Booting pool A_POOL with kernel /ptmp/jgh/ukernel
02/04/94 16:45:49 - mppstart: Booting pool A_POOL with PAL /mpp/os/maxpal
02/04/94 16:45:49 - mppstart: Booting I/O nodes with /mpp/os/iog_os
02/04/94 16:45:51 - Channel/Gateway 0 has been enabled.
```

```
02/04/94 16:45:51 - Channel/Gateway 1 has been enabled.
02/04/94 16:45:51 - Channel/Gateway 2 has been enabled.
02/04/94 16:45:51 - Channel/Gateway 3 has been enabled.
02/04/94 16:45:51 - All pools marked.
02/04/94 16:45:51 -       MPP_UNAVAILABLE   set.
02/04/94 16:45:51 - Administrative pool database invalidated in Unicos kernel.
02/04/94 16:45:51 - Barrier database invalidated in Unicos kernel.
02/04/94 16:45:51 - MPP configuration set to 8:4:4, with 4 redundant nodes.
02/04/94 16:45:51 - MPP barrier information has been loaded.
02/04/94 16:45:51 - MPP redundant node mapping table has been loaded.
02/04/94 16:45:51 - Configuration driver statistics:
02/04/94 16:45:51 -   Successful allocations: 1
02/04/94 16:45:51 -       OS partitions: 0    HW partitions: 1
02/04/94 16:45:51 -       Successful interactive allocations: 1
02/04/94 16:45:51 -   Failed allocations: 2
02/04/94 16:45:51 -       Unable to get barrier resources: 67
02/04/94 16:45:51 -       Unable to get node resources: -65
02/04/94 16:45:51 - MPP Administrative Pools have been loaded.
02/04/94 16:45:51 - mppd: Boot directory is /ptmp/jgh/os/cmd/mppadmin
02/04/94 16:45:51 - mppstart: MPP boot sequence completed
02/04/94 16:46:00 - mppd: sanity: All PEs are responding.
02/04/94 16:47:29 - Partition 2 has been allocated.
02/04/94 16:47:29 -       Application name : ft.
02/04/94 16:47:29 -       Uid : root (0) Owning process id : 51718
02/04/94 16:47:29 -       Partition type :Hardware
02/04/94 16:47:29 -       Barrier circuit 0 allocated, mask = 0x1111
02/04/94 16:47:29 -       Barrier bypass: PE 0x0 snc 0x2000
02/04/94 16:47:29 -       Node count:16
02/04/94 16:47:29 -       Nodes in partition :
02/04/94 16:47:29 -       0x000 0x002 0x004 0x006 0x010 0x012 0x014 0x016
02/04/94 16:47:29 -       0x100 0x102 0x104 0x106 0x110 0x112 0x114 0x116
02/04/94 16:47:29 - Partition 7 has been allocated.
02/04/94 16:47:29 -       Application name : ft.
02/04/94 16:47:29 -       Uid : root (0) Owning process id : 51720
02/04/94 16:47:29 -       Partition type :Hardware
02/04/94 16:47:29 -       Barrier circuit 0 allocated, mask = 0x1111
02/04/94 16:47:29 -       Barrier bypass: PE 0xc snc 0x2000
02/04/94 16:47:29 -       Node count:16
02/04/94 16:47:29 -       Nodes in partition :
02/04/94 16:47:29 -       0x008 0x00a 0x00c 0x00e 0x018 0x01a 0x01c 0x01e
02/04/94 16:47:29 -       0x108 0x10a 0x10c 0x10e 0x118 0x11a 0x11c 0x11e
02/04/94 16:47:29 - Partition 18 has been allocated.
02/04/94 16:47:29 -       Application name : ft.
02/04/94 16:47:29 -       Uid : root (0) Owning process id : 51719
02/04/94 16:47:29 -       Partition type :Hardware
02/04/94 16:47:29 -       Barrier circuit 0 allocated, mask = 0x1111
02/04/94 16:47:29 -       Barrier bypass: PE 0x30 snc 0x2000
02/04/94 16:47:29 -       Node count:16
02/04/94 16:47:29 -       Nodes in partition :
02/04/94 16:47:29 -       0x020 0x022 0x024 0x026 0x030 0x032 0x034 0x036
02/04/94 16:47:29 -       0x120 0x122 0x124 0x126 0x130 0x132 0x134 0x136
02/04/94 16:47:29 - mppd: sanity: All PEs are responding.
02/04/94 16:47:29 - Partition 18 Agent Notice: Agent running
02/04/94 16:47:29 - Partition 2 Agent Notice: Agent running
02/04/94 16:47:29 - Partition 7 Agent Notice: Agent running
02/04/94 16:47:39 - Partition 18 Agent Notice: exiting normally
02/04/94 16:47:39 - Partition 2 Agent Notice: exiting normally
02/04/94 16:47:39 - Partition 18 is released.
```

```
02/04/94 16:47:39 - Partition 7 Agent Notice: exiting normally
02/04/94 16:47:39 - Partition 2 is released.
02/04/94 16:47:39 - Partition 7 is released.
```

Example 2: A sample CRAY T3D system daemon log (`mppd.log`) appears as follows:

```
02/04/94 14:05:34 - MPPD_DISABLE_GATEWAY (0) request succeeded
02/04/94 14:05:34 - MPPD_DISABLE_GATEWAY (1) request succeeded
02/04/94 15:05:34 - Gateway 2 not configured
02/04/94 15:05:34 - Gateway 3 not configured
02/04/94 15:05:41 - Cmd request received: type 2
02/04/94 15:05:41 - Loading configuration file /typhoon/u59/piatz/sn6001
mppd: Warning: I/O node C08 is unavailable
mppd: Warning: I/O node C18 is unavailable
02/04/94 15:16:18 - ERROR request received: type 4
02/04/94 15:16:18 - killing application: pid = 2400 signo = 26
```

FILES

`/usr/spool/mpp/`	Directory in which the CRAY T3D system log files are created and kept

SEE ALSO

`mppd(8)`

NAME

mppboot – Configures and boots CRAY T3D systems

SYNOPSIS

mppboot [-c *config_file*] [-f *trace_flags*] [-g *gateway_device*] [-m *message_level*] [-p] [-r *route_file*]

mppboot [-c *config_file*] [-f *trace_flags*] [-g *gateway_device*] [-m *message_level*] [-p] [-R *route_file*]

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The mppboot command configures and boots CRAY T3D systems. It executes the mpproute(8) command to generate route tables, if necessary, deadstarts the MPP using mppstart(8), starts the MPP daemon process (mppd(8)), if necessary, and initializes the configuration driver.

The mppboot command accepts the following options:

-c *config_file* Specifies the MPP configuration file. If the -c option is specified, a new route table file is generated using the specified configuration file.

-f *trace_flags* Specifies a trace mask to be loaded with the OS support PAL binary (maxpal).

-g *gateway_device* Specifies the I/O gateway device to be used as the deadstart device. Any configured I/O gateway device can be used to deadstart the CRAY T3D system. If no device is specified, mppstart chooses one of the configured gateways.

-m *message_level* Specifies the level of informational messages to be output during the deadstart sequence. Valid levels include the following:

0 Silent; error messages only. (Default)

1 Trace; packet headers written to standard output.

2 Debug; formatted packets written to standard output.

3 Raw; unformatted packets written to standard output.

-p Partial boot. The -p option can be used to boot only the primary boot PAL binary (admpal) on all the nodes. This preserves memory contents on the processing elements (PEs) and allows the system to be dumped.

-r *route_file* Specifies the routing file used to boot. The default is the mpp.route file in the current working directory. Using the -r option causes the modification time of the configuration file to be checked against that of the specified route file. If the configuration file has been modified since the route file was created, a new route file is generated.

-R *route_file* Same as -r option, except that the route table file is not regenerated, even if the configuration file has been modified.

EXAMPLES

Sample output from the mppboot command is as follows:

```
# /mpp/bin/mppboot -c /mpp/cf/sn6001
mppboot : Starting the MPP daemon ...
mppboot : Disabling all administrative resource pools...
mppboot : Generating route tables ...
mppboot : /mpp/bin/mpproute -c /mpp/cf/sn6001 -r ./mpp.route
mppboot : /mpp/bin/mppstart -r ./mpp.route
mppstart: Killing all MPP applications...
mppstart: Issuing Master Clear on /dev/mpp/iog00
mppstart: Device /dev/mpp/iog00 is I/O node 0xc30
mppstart: Issuing Master Clear on /dev/mpp/iog01
mppstart: Device /dev/mpp/iog01 is I/O node 0xc3e
mppstart: Issuing Master Clear on /dev/mpp/iog02
mppstart: Device /dev/mpp/iog02 is I/O node 0xc18
mppstart: Issuing Master Clear on /dev/mpp/iog03
mppstart: Device /dev/mpp/iog03 is I/O node 0xc16
mppstart: Deadstart node is 0xc30, SROM version 2064
mppstart: Booting admpal /mpp/os/admpal
mppstart: Booting pool A_POOL with kernel /mpp/os/ukernel
mppstart: Booting pool A_POOL with PAL /mpp/os/maxpal
mppstart: Booting I/O nodes with /mpp/os/iog_os
mppstart: Initializing HISP on gateway device /dev/mpp/iog00, mode 3
mppstart: Initializing HISP on gateway device /dev/mpp/iog01, mode 3
mppstart: Initializing HISP on gateway device /dev/mpp/iog02, mode 3
mppstart: Initializing HISP on gateway device /dev/mpp/iog03, mode 3
mppstart: Initializing administrative resource pools...
```

SEE ALSO

mppcmd(8), mppd(8), mpproute(8), mppstart(8)

NAME

mppcmd – Sends a request to the CRAY T3D daemon

SYNOPSIS

mppcmd *command option arg* [*arg* ...]

mppcmd enable gateway *gateway_dev* [*gateway_dev* ...]
mppcmd disable gateway *gateway_dev* [*gateway_dev* ...]
mppcmd enable node *node* [*node* ...]
mppcmd disable node *node* [*node* ...]
mppcmd enable pool *pool_id* [*pool_id* ...]
mppcmd disable pool *pool_id* [*pool_id* ...]
mppcmd kill node *signo node* [*node* ...]
mppcmd kill partition *signo partition_id* [*partition_id* ...]
mppcmd load *config_file*
mppcmd set *poolid attribute* [*attribute* ...]
mppcmd clear *poolid attribute* [*attribute* ...]
mppcmd shutdown *grace_period*

Deferred implementation:
mppcmd boot *node* [*node* ...]
mppcmd halt *node* [*node* ...]
mppcmd replace *node* [*node* ...]

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The mppcmd command issues requests to the CRAY T3D daemon (mppd(8)) on behalf of a user. If no command-line arguments are specified, mppcmd enters an interactive loop. Subcommands can then be entered after the mppcmd prompt. The interactive session can be terminated by entering the quit subcommand.

All subcommands require that the CRAY T3D daemon (mppd(8)) be running. Any failures are logged in the daemon log and reported back to the requestor through a response pipe.

The following mppcmd subcommands are available:

enable gateway *gateway_dev* [*gateway_dev* ...]
disable gateway *gateway_dev* [*gateway_dev* ...]
: Argument taken is a list of path names of the devices to be enabled or disabled. The daemon then opens the device and makes the necessary ioctl() requests to enable or disable the gateways.

enable node *node* [*node* ...]
disable node *node* [*node* ...]
: Argument taken is a list of nodes to be enabled or disabled. The daemon makes a request to the configuration driver to enable or disable these nodes.

`enable pool` *poolid* [*poolid* [*poolid* ...]]
`disable pool` *poolid* [*poolid* [*poolid* ...]]

Argument taken is a list of pool IDs to enable or disable. A pool ID of `all` indicates that all pools should be enabled or disabled. The daemon makes a request to the configuration driver to enable or disable these pools.

`kill node` *signo node* [*node* ...]

Argument taken is a physical node number(s). The daemon queries the configuration code and searches for the partition(s) containing the specified physical node(s). The daemon then sends the specified signal to the application(s) running in the partition(s).

`kill partition` *signo partition_id* [*partition_id* ...]

Argument taken is a list of partition IDs. The daemon queries the configuration code and searches for the partitions with the specified IDs. The daemon then sends the specified signal to the applications running in those partitions.

`load` *config_file*

Argument taken is a path name for the new configuration file. The daemon makes the calls to the configuration driver to activate this configuration. This process fails if there are any active pools or partitions.

`set` *poolid attribute* [*attribute* ...]
`clear` *poolid attribute* [*attribute* ...]

Argument taken is a list of attributes to be set or cleared on the pool identified by *poolid*. A pool ID of `all` indicates that all pools should be updated. Valid attributes are: group ID numbers, `BATCH`, `INTERACTIVE`, `BOTH` (batch and interactive), `AVAILABLE`, and `UNAVAILABLE`.

`shutdown` *grace_period*

Argument taken is a flag indicating the delay (in seconds) before shutdown. If a nonzero delay is specified, the daemon first disables all pools to halt incoming requests and then sends each active partition a `SIGSHUTDN` signal. The daemon then waits the specified delay before killing each application and resetting the CRAY T3D system to a known state.
NOTE: For an orderly shutdown of the CRAY T3D system and the Cray host system, you can put the call to `mppcmd shutdown` *grace_period* into the normal UNICOS shutdown script.

`boot` *node* [*node* ...]

NOTE: Deferred implementation.
Argument taken is a list of physical nodes to be rebooted. The daemon downloads new route tables, the microkernel binary, and the operating system support PAL to each processing element (PE) on the specified nodes.

`halt` *node* [*node* ...]

NOTE: Deferred implementation.
Argument taken is a list of physical nodes to be halted. The daemon sends a hardware IPC message to each PE on the specified nodes, indicating that they should perform a software reset.

`replace` *node* [*node* ...]

NOTE: Deferred implementation.
Argument taken is a list of physical nodes to be mapped out. The daemon issues a request to the configuration driver to map in a redundant node to replace the bad node. If there is an available redundant node, the daemon updates route tables on all nodes in the system. If there are no available redundant nodes, the bad nodes are disabled.

SEE ALSO

`mppd(8)`

NAME

mppd – Starts the CRAY T3D daemon

SYNOPSIS

mppd [-l *logfile*] [-p]

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The CRAY T3D daemon is a multitasked process. It includes the request processor task, the error logger task, and the partition cleanup task.

The request processor task handles any user request initiated by sending a request over the named pipe /usr/spool/mpp/mppd.reqpipe. It also handles internal request originating from the error logger task through a shared memory mechanism.

The error logger task monitors and logs all CRAY T3D system activity to the CRAY T3D system log (mppsyslog), CRAY Y-MP system console, and the diagnostics daemon (dgdemon(8)), when appropriate. It also alerts the request processor task to take any necessary action in the event of some catastrophic error coming from the CRAY T3D system.

The partition cleanup task ensures that all partitions are freed once the agent associated with that partition exits.

The mppd command accepts the following options:

-l *logfile* Specifies a log file to which the daemon writes its internal daemon log messages. The default is the /usr/spool/mpp/mppd.log file.

-p Physically lock (plock) the daemon in memory.

Request Processor Task

This mppd task handles all requests. Requests can be generated through a command interface or by the error logger task. The command interface (mppcmd(8)) to the processor request task is through a named pipe. The mppcmd(8) command writes the request structure defined below to the pipe and waits for a response on a response pipe created for the request. The error logger task interface to the request task is through shared memory. The error logger task writes the request structure to a circular buffer in shared memory and signals the request processor that there is work to be done.

The mppd command request processor header structure is as follows:

```
typedef struct {
      uint magic;
      uint type;
      int requestor_pid;
} mppd_cmdreq_t;
```

The mppd command response structure is as follows:

```
#define STATUS_OK 0
#define STATUS_FAIL -1
struct mppd_response {
      uint status:32;
      uint code  :32;
};
```

The following subsections define the possible request types and their associated data structures.

Shut down the CRAY T3D system

Argument taken is a flag indicating the delay before shutdown. If a nonzero delay is specified, the daemon first disables all pools to halt incoming requests and then sends each active partition a SIGSHUTDN signal. It then waits the specified delay before killing each application and resetting the CRAY T3D system to a known state.

The mppd shutdown request structure is as follows:

```
struct mppd_shutdown_req {
      mppd_cmdreq_t hdr;
      int delay;
};
```

Kill application by mppexec PID

Argument taken is the process ID (PID) of an application to be killed. The daemon sends the specified signal to the mppexec process associated with that PID.

Kill application by PE or partition ID

Argument taken is a physical processing element (PE) number or partition ID. The daemon queries the configuration code search for the partition containing this physical PE or having the specified ID. The daemon then attempts to kill the application. Any failures are logged in the daemon log and reported back to the requestor through the response pipe.

The mppd kill request structure is as follows:

```
/*
 * MPPD_KILL_BYPID request
 * MPPD_KILL_BYPART request
 * MPPD_KILL_BYNODE request
 */
struct mppd_kill_req {
      mppd_cmdreq_t hdr;
      int signo;        /* signal to send */
      int id[MPPD_MAX_ARGS];  /* pid's, partid's, or nodes */
};
```

Reconfigure the CRAY T3D system

Argument taken is a path name for the new configuration file. The daemon makes the calls to the configuration driver to activate this configuration. This process fails if there are any active pools or partitions. Any failures are logged in the daemon log and reported back to the requestor through the response pipe.

The mppd load pool request structure is as follows:

```
struct mppd_ldpool_req {
      mppd_cmdreq_t hdr;
      char cffile[FILENAME_MAX];  /* path to config file */
};
```

Enable or disable pool

Argument taken is a pool to be enabled or disabled. The daemon makes a request to the configuration driver to disable this pool. Any failure is logged in the daemon log and reported back to the requestor through the response pipe.

The mppd pool request structure is as follows:

```
/*
 * MPPD_DISABLE_POOL request
 * MPPD_ENABLE_POOL request
 * MPPD_SETPOOL_ATTR request
 * MPPD_CLRPOOL_ATTR request
 */
struct mppd_pool_req {
      mppd_cmdreq_t hdr;
      uint poolid;
      uint attributes;
};
```

Enable or disable gateway

Argument taken is the path name of the device to be enabled. The daemon then opens the device and makes an `ioctl()` request to enable the gateway. If the device specified is nonvalid or if the enable failed, the requestor is notified of the failure through the response pipe.

The mppd gateway request structure is as follows:

```
/*
 * MPPD_DISABLE_GATEWAY request
 * MPPD_ENABLE_GATEWAY request
 */
struct mppd_gateway_req {
      mppd_cmdreq_t hdr;
      uint gateways;
};
```

Enable or disable node

Argument taken is a bitlist indicating which nodes are to be enabled or disabled. The daemon makes a request to the configuration driver to disable these nodes.

Halt node

Argument taken is a bitlist indicating which physical nodes are to be halted. The daemon sends a hardware IPC message to each PE on the specified nodes indicating that they should perform a software reset.

Reboot node

Argument taken is a bitlist indicating which physical nodes are to be rebooted. The daemon downloads new route tables, the microkernel binary, and then the operating system support PAL.

Map out bad node

Argument taken is a bitlist indicating which physical nodes are to be mapped out. The daemon issues a request to the configuration driver to map in a redundant node to replace the bad node. If there is an available redundant node, the daemon updates route tables on all nodes in the system. If there are no available redundant nodes, the bad node(s) are disabled. Any failures are logged in the daemon log and reported back to the requestor through the response pipe.

The mppd node request structure is as follows:

```
/*
 * MPPD_DISABLE_NODE request
 * MPPD_ENABLE_NODE request
 * MPPD_PE_HALT request
 * MPPD_PE_REBOOT request
 * MPPD_MAP_REDUNDANT request
 */
struct mppd_node_req {
      mppd_cmdreq_t hdr;
      BitVector nodemask;     /* bitlist indicating nodes */
};
```

Error Logger Task

The error logger task handles all logging and error packet processing. It sleeps on a read from the /dev/mpp/mpplog device, waiting for a message to process and log. It is capable of generating requests to the request processor task on a few special error indications. The main purpose of this task is to log information about the functioning of the CRAY T3D support software and the CRAY T3D hardware. This includes the I/O channels connecting the CRAY T3D system with the CRAY Y-MP system. A special socket connection to the UNICOS dgdemon(8) process allows indications of CRAY T3D hardware errors to be handled automatically.

For each message received, the error logger creates an entry in the CRAY T3D system log. For more information about error logging, error messages, and handling errors on CRAY T3D systems, see the *CRAY T3D Administrator's Guide*, publication SG–2507.

Partition Cleanup Task

The partition cleanup task handles the cleanup of all partitions once the mppexec(1) process has exited. It sleeps in the configuration kernel code, waiting for a partition to go to a ZOMBIE state. The task then sends a global exit request to each PE in the partition. The PE then cleans up the threads and tasks, resets hardware as needed, and waits for the next user. Once all PEs have responded successfully, the task notifies the configuration driver that the partition resources are available again for a new application.

SEE ALSO

mppexec(1), mppcmd(8)

dgdemon(8) in the *UNICOS Administrator Commands Reference Manual*, publication SR–2022

CRAY T3D Administrator's Guide, publication SG–2507

NAME

mppping – Tests the CRAY T3D gateway connections and compute processing elements (PEs)

SYNOPSIS

mppping [-g] [-n *n_times*] [-p] [-s *pktsize*] [-v] [*gateway_device* [*gateway_device*]]

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The mppping command attempts to send an echo packet to both the input and output sides of each of the specified gateways to see if the gateway responds. It also determines which compute processing elements (PEs) are up and running.

If no gateways are specified, then mppping sends echo packets to all enabled gateways. A gateway must be enabled in order to send the echo packets.

If at least one enabled gateway responds, then mppping has the I/O gateway read a word of memory on each configured compute PE and send the word of memory back to the mppping command. The mppping command then interprets the memory to determine which PEs are up and running.

To test only the I/O gateways, use the -g option. To interrogate only the compute PEs, use the -p option.

The mppping command accepts the following options:

-g Tests only the I/O gateways.

-n *n_times* For each gateway, sends the packet *n_times*. The default is 1.

-p Interrogates only the compute PEs.

-s *pktsize* Sends packets that are *pktsize* bytes long. The default is 256 bytes.

-v (Verbose) Lists all configured gateways and whether they are enabled or disabled. Writes an entry for each configured compute PE, indicating whether the microkernel on that PE is up or down, and the state of the PE. Valid PE states are as follows:

idle	PE is idle.
halted	PE has halted.
booting	PE is being booted.
user init	User is being downloaded.
user startup	User thread is being started.
user running	User is running.
user exit	User is exiting.

gateway_device Path name of the device special file.

EXAMPLES

Example 1: The default mpppping output is as follows:

```
# mpppping
Gateway /dev/mpp/iog01 output node responding
Gateway /dev/mpp/iog01  input node responding
All PEs are UP
```

Example 2: To test only the I/O gateways:

```
# mpppping -g
Gateway /dev/mpp/iog01 output node responding
Gateway /dev/mpp/iog01  input node responding
```

Example 3: To send a packet *n* number of times:

```
# mpppping -n 3
Gateway /dev/mpp/iog01 output node responding
Gateway /dev/mpp/iog01  input node responding
Gateway /dev/mpp/iog01 output node responding
Gateway /dev/mpp/iog01  input node responding
Gateway /dev/mpp/iog01 output node responding
Gateway /dev/mpp/iog01  input node responding
All PEs are UP
```

Example 4: To interrogate only the compute PEs:

```
# mpppping -p
All PEs are UP
```

Example 5: To send packets of a specified size (default is 256):

```
# mpppping -s 256
Gateway /dev/mpp/iog01 output node responding
Gateway /dev/mpp/iog01  input node responding
All PEs are UP
```

Example 6: To list all configured gatways and whether they are enabled or disabled:

```
# mpppping -v
Gateway /dev/mpp/iog01 configured, enabled
Gateway /dev/mpp/iog02 configured, enabled
Gateway /dev/mpp/iog03 configured, enabled

Gateway /dev/mpp/iog00 output node responding
Gateway /dev/mpp/iog00  input node responding
Gateway /dev/mpp/iog01 output node responding
Gateway /dev/mpp/iog01  input node responding
Gateway /dev/mpp/iog02 output node responding
Gateway /dev/mpp/iog02  input node responding
Gateway /dev/mpp/iog03 output node responding
Gateway /dev/mpp/iog03  input node responding

Node 0x000  PE 0 (0x000) status: UP   state: user running
            PE 1 (0x001) status: UP   state: user running
Node 0x002  PE 0 (0x002) status: UP   state: user running
```

```
             PE 1 (0x003) status: UP   state: user running
Node 0x004   PE 0 (0x004) status: UP   state: user running
             PE 1 (0x005) status: UP   state: user running
Node 0x006   PE 0 (0x006) status: UP   state: user running
             PE 1 (0x007) status: UP   state: user running
Node 0x008   PE 0 (0x008) status: UP   state: user running
             PE 1 (0x009) status: UP   state: user running
Node 0x00a   PE 0 (0x00a) status: UP   state: user running
             PE 1 (0x00b) status: UP   state: user running
Node 0x00c   PE 0 (0x00c) status: UP   state: user running
             PE 1 (0x00d) status: UP   state: user running
Node 0x00e   PE 0 (0x00e) status: UP   state: user running
             PE 1 (0x00f) status: UP   state: user running
Node 0x010   PE 0 (0x010) status: UP   state: user running
             PE 1 (0x011) status: UP   state: user running
Node 0x012   PE 0 (0x012) status: UP   state: user running
             PE 1 (0x013) status: UP   state: local exit
Node 0x014   PE 0 (0x014) status: UP   state: user exit
             PE 1 (0x015) status: UP   state: user exit
Node 0x016   PE 0 (0x016) status: UP   state: user exit
             PE 1 (0x017) status: UP   state: user exit
Node 0x018   PE 0 (0x018) status: UP   state: user exit
             PE 1 (0x019) status: UP   state: user exit
Node 0x01a   PE 0 (0x01a) status: UP   state: user exit
             PE 1 (0x01b) status: UP   state: user exit
Node 0x01c   PE 0 (0x01c) status: UP   state: user exit
             PE 1 (0x01d) status: UP   state: user exit
Node 0x01e   PE 0 (0x01e) status: UP   state: user exit
             PE 1 (0x01f) status: UP   state: user exit
Node 0x020   PE 0 (0x020) status: UP   state: user exit
             PE 1 (0x021) status: UP   state: user exit
Node 0x022   PE 0 (0x022) status: UP   state: user exit
             PE 1 (0x023) status: UP   state: user exit
Node 0x024   PE 0 (0x024) status: UP   state: user exit
             PE 1 (0x025) status: UP   state: user exit
Node 0x026   PE 0 (0x026) status: UP   state: user exit
             PE 1 (0x027) status: UP   state: user exit
Node 0x028   PE 0 (0x028) status: UP   state: idle
             PE 1 (0x029) status: UP   state: idle
Node 0x02a   PE 0 (0x02a) status: UP   state: idle
             PE 1 (0x02b) status: UP   state: idle
                          .
                          .
                          .
Node 0x33c   PE 0 (0x33c) status: UP   state: idle
             PE 1 (0x33d) status: UP   state: idle
Node 0x33e   PE 0 (0x33e) status: UP   state: idle
             PE 1 (0x33f) status: UP   state: idle
All PEs are UP
```

NAME

mpproute – Generates CRAY T3D binary configuration file with routing tables

SYNOPSIS

mpproute [-c *config_file*] [-r *route_file*]

mpproute -i [-s *source* [-d *destination*]] [-r *route_file*]

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The mpproute command is used to generate routing tables for CRAY T3D systems. The mpproute command looks for a file named mppconfig.local, which contains information regarding system configuration, bad nodes, and downed links within the network, and generates a binary routing table file named mpp.route. The mpp.route file contains a unique routing table for each node in the system. The routing table is used as input to the mppstart(8) command. The mppstart command downloads the routing tables to the appropriate nodes in the system.

The CRAY T3D configuration file (mppconfig.local) must be created by the administrator in order to recover from system component failures. The format of the CRAY T3D configuration file is described in mppconfig(5).

If the -r option is specified, the routing file is created in, or read from, the specified path. If the -r option is not specified, the mpp.route file in the current working directory is assumed.

The -i option can be used to obtain information from a previously created routing table file. In this case, mpproute reads an existing mpp.route file and displays the configuration of the system for which the file was generated, the path to the CRAY T3D configuration file used to generate the file, and the routing tables themselves.

The -s and -d options can be used to control what information is displayed. If the -s option is specified, only the routing table for the specified source node is displayed. If the -d option is also specified, only the route from the source node to the destination node is displayed.

The configuration of the CRAY T3D system is obtained from the CRAY T3D configuration file.

The mpproute command accepts the following options:

-c *config_file* Specifies the CRAY T3D configuration file. The default is the mppconfig.local file in the current working directory.

-d *destination* When used in conjunction with the -s option, the -d option displays routing information for a particular route from the specified source node to the specifed destination node. This option is only valid when used in conjuction with the -i and -s options.

-i Specifies that an existing mpp.route file is to be read and the information formatted and displayed to the terminal.

-r *route_file* Specifies the binary routing table file. The default is the mpp.route file in the current working directory.

-s *source* Displays the routing table information for the specified node. This option is only valid when used in conjuction with the -i option.

EXAMPLES

The following example shows how to check the routing tables for a single PE at node number 0x000:

```
# mpproute -i -s 0x000

Cray T3D Routing Table Generation Utility Version 2.1
           4x4x2 non-integrated chassis

Configuration File: /mpp/cf/sn6202

I/O Gateway Placement
Physical Logical   X+    X-    Z+    Z-
-------- ------- ----- ----- ----- -----
 0xc20    0x4a0  0x002 0x000 0x100 0x000
 0xc02    0x482  0x114 0x112 0x012 0x112

Redundant Node Placement
Physical Logical   X+    X-    Y+    Y-    Z+    Z-   Mapping
-------- ------- ----- ----- ----- ----- ----- -----  -------
 0x906    0x906  0x100 0x106 0x916 0x936 0x006 0x106
 0x916    0x916  0x110 0x116 0x926 0x906 0x016 0x116
 0x926    0x926  0x120 0x126 0x936 0x916 0x026 0x126
 0x936    0x936  0x130 0x136 0x906 0x926 0x036 0x136

Routing table for node -> 0x000 (0)

Logical     Physical    Tag          dZ(vc),dY(vc),dX(vc)
-------     --------   -------       --------------------
 0x000        0x000         4040     0(0),  0(1),  0(1)
 0x002        0x002           55     0(0),  0(0),  3(1)
 0x004        0x004       404014     0(1),  0(1),  4(0)
 0x006        0x006           77     0(0),  0(0), -1(1)
 0x010        0x010         5700     0(0),  1(1),  0(0)
 0x012        0x012         5755     0(0),  1(1),  3(1)
 0x014        0x014         5714     0(0),  1(1),  4(0)
 0x016        0x016         5777     0(0),  1(1), -1(1)
 0x020        0x020         1600     0(0),  2(0),  0(0)
 0x022        0x022         1655     0(0),  2(0),  3(1)
 0x024        0x024         1614     0(0),  2(0),  4(0)
 0x026        0x026         1677     0(0),  2(0), -1(1)
 0x030        0x030         7700     0(0), -1(1),  0(0)
 0x032        0x032         7755     0(0), -1(1),  3(1)
 0x034        0x034         7714     0(0), -1(1),  4(0)
 0x036        0x036         7777     0(0), -1(1), -1(1)
 0x100        0x100       550000     3(1),  0(0),  0(0)
 0x102        0x102       570055     1(1),  0(0),  3(1)
 0x104        0x104       570014     1(1),  0(0),  4(0)
 0x106        0x106       570077     1(1),  0(0), -1(1)
 0x110        0x110       575700     1(1),  1(1),  0(0)
 0x112        0x112       575755     1(1),  1(1),  3(1)
 0x114        0x114       575714     1(1),  1(1),  4(0)
 0x116        0x116       575777     1(1),  1(1), -1(1)
 0x120        0x120       571600     1(1),  2(0),  0(0)
```

```
0x122       0x122       571655    1(1),  2(0),  3(1)
0x124       0x124       571614    1(1),  2(0),  4(0)
0x126       0x126       571677    1(1),  2(0), -1(1)
0x130       0x130       577700    1(1), -1(1),  0(0)
0x132       0x132       577755    1(1), -1(1),  3(1)
0x134       0x134       577714    1(1), -1(1),  4(0)
0x136       0x136       577777    1(1), -1(1), -1(1)
0xc20       0xc20       170000    1(0),  0(0),  0(0)
0xc30       0xc30       160000    2(0),  0(0),  0(0)
0xc02       0xc02       565755    2(1),  1(1),  3(1)
0xc12       0xc12       555755    3(1),  1(1),  3(1)
0x906       0x906       560077    2(1),  0(0), -1(1)
0x916       0x916       565777    2(1),  1(1), -1(1)
0x926       0x926       561677    2(1),  2(0), -1(1)
0x936       0x936       567777    2(1), -1(1), -1(1)
```

SEE ALSO

mppconfig(5), mppstart(8)

NAME

mppstart – Initiates the CRAY T3D system deadstart sequence

SYNOPSIS

mppstart [-f *trace_flags*] [-g *gateway_device*] [-m *message_level*] [-p] [-r *route_file*]

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The mppstart command downloads the CRAY T3D system software and starts the CPUs at each node in the CRAY T3D system from the CRAY Y-MP system.

The mppstart command reads the routing table file to obtain information regarding the desired system characteristics. It then issues a master clear function over the LOSP channel to the deadstart node of the CRAY T3D system and downloads, to the deadstart node, a copy of the primary boot Privileged Architecture Library (PAL), routing tables, I/O node control software, system PAL, and microkernel binary. The deadstart node then propagates these binaries to the appropriate nodes of the CRAY T3D system.

By default, mppstart looks in the current working directory for the following file:

mpp.route File generated by mpproute(8), containing routing and configuration information

Then mppstart looks in the /mpp/os directory for the following files:

admpal Primary boot PAL binary

iog_os I/O node control software binary

maxpal OS support PAL binary

ukernel Microkernel binary

The -r option can be used to specify an alternate routing table file. If no routing table file is specified, mppstart will look in the current working directory for a file named mpp.route.

The mppstart command accepts the following options:

-f *trace_flags* Specifies a trace mask to be loaded with the maxpal binary.

-g *gateway_device* Specifies the I/O gateway device to be used as the deadstart device. Any configured I/O gateway device can be used to deadstart the CRAY T3D system. If no device is specified, mppstart will choose one of the configured gateways.

-m *message_level* Specifies the level of informational messages to be output during the deadstart sequence. Valid levels include the following:

0 Silent; error messages only (Default)

1 Trace; packet headers written to standard output

2 Debug; formatted packets written to standard output

3 Raw; unformatted packets written to standard output

-p Partial boot. The -p option can be used to boot only the admpal on all the nodes. This will preserve memory contents on the processing elements (PEs) and allow the system to be dumped.

`-r` *route_file* Specifies the routing file with which to boot. The default is the `mpp.route` file in the current working directory.

SEE ALSO

`mpproute`(8)

NAME

mppstat – Displays CRAY T3D system resource status

SYNOPSIS

```
mppstat -a
mppstat [-b] [-d] [-m] [-p] [-P] [-s]
```

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The mppstat command can be used to view CRAY T3D system resource allocation information.

The mppstat command accepts the following options:

-a Same as specifying all other options.

-b Displays the status of barrier wires that are allocated and/or bad. This shows the points within the barrier network that are allocated for all active partitions or turned off due to corresponding disabled nodes.

-d Displays the current list of disabled nodes. Nodes are disabled automatically when a microkernel panic occurs or when the hardware or software fails to respond to the "sanity check" message that is issued regularly. They also can be disabled manually via mppcmd(8).

-m Displays the memory size of each processing element (PE) in the CRAY T3D system.

-p Displays information on active and queued user partitions only. The displayed order of the partitions reflects the order in which they will be processed, aside from considerations relating to the ExpressTime and MaxWaitTime tuning parameters, and node/PE availability.

-P Displays information on administrative resource pool usage, such as nodes allocated to each pool, disabled nodes in the pool, number of active partitions in the pool, tuning parameter values, and pool state.

-s Displays CRAY T3D configuration driver statistics.

If no options are specified, a high-level profile of the CRAY T3D system configuration is displayed.

EXAMPLES

To display information about all aspects of CRAY T3D resource status, enter the following:

```
# mppstat -a
Configuration Information:

      Torus PE dimensions :   16 x 4 x 4
      Redundant PEs :   8 (total) 0 (mapped in)

      Maximum pools :         7     Pools in use :         1
      Maximum partitions :    16    Partitions in use :    3
      Total PEs available :   160
      Disabled PE count :     0
      Barrier Initialized?    yes   Pools Initialized?     yes
      Config Time :           Fri Feb  4 16:45:51 1994

Received information on 1 Pool[s]
```

```
   Pool 0 - A_POOL:
      Attributes :  Available Batch Interactive
      Flags :
      Gids :  os(1013)
      PE Member Count : 256
      Available PEs  : 160
      Pool PE Shape : 16x4x4
      Active/Zombie partitions : 3  Maximum allowed : 16
      Express time limit for jobs in pool :     0 second(s)
      Maximum wait time for jobs in pool :      0 second(s)
      Nodes in Pool :
        0x000 0x002 0x004 0x006 0x008 0x00a 0x00c 0x00e
        0x010 0x012 0x014 0x016 0x018 0x01a 0x01c 0x01e
        0x020 0x022 0x024 0x026 0x028 0x02a 0x02c 0x02e
        0x030 0x032 0x034 0x036 0x038 0x03a 0x03c 0x03e
        0x100 0x102 0x104 0x106 0x108 0x10a 0x10c 0x10e
        0x110 0x112 0x114 0x116 0x118 0x11a 0x11c 0x11e
        0x120 0x122 0x124 0x126 0x128 0x12a 0x12c 0x12e
        0x130 0x132 0x134 0x136 0x138 0x13a 0x13c 0x13e
        0x200 0x202 0x204 0x206 0x208 0x20a 0x20c 0x20e
        0x210 0x212 0x214 0x216 0x218 0x21a 0x21c 0x21e
        0x220 0x222 0x224 0x226 0x228 0x22a 0x22c 0x22e
        0x230 0x232 0x234 0x236 0x238 0x23a 0x23c 0x23e
        0x300 0x302 0x304 0x306 0x308 0x30a 0x30c 0x30e
        0x310 0x312 0x314 0x316 0x318 0x31a 0x31c 0x31e
        0x320 0x322 0x324 0x326 0x328 0x32a 0x32c 0x32e
        0x330 0x332 0x334 0x336 0x338 0x33a 0x33c 0x33e
      Available nodes in Pool :
        0x000 0x002 0x004 0x006 0x010 0x012 0x014 0x016
        0x100 0x102 0x104 0x106 0x110 0x112 0x114 0x116
        0x200 0x202 0x204 0x206 0x208 0x20a 0x20c 0x20e
        0x210 0x212 0x214 0x216 0x218 0x21a 0x21c 0x21e
        0x220 0x222 0x224 0x226 0x228 0x22a 0x22c 0x22e
        0x230 0x232 0x234 0x236 0x238 0x23a 0x23c 0x23e
        0x300 0x302 0x304 0x306 0x308 0x30a 0x30c 0x30e
        0x310 0x312 0x314 0x316 0x318 0x31a 0x31c 0x31e
        0x320 0x322 0x324 0x326 0x328 0x32a 0x32c 0x32e
        0x330 0x332 0x334 0x336 0x338 0x33a 0x33c 0x33e

Current list of disabled nodes:


Received information on 3 Partition[s]

   Partition 8:
      State : Active    Type :      Hardware
      Flags :
      Owner :     root (0)    Owning process: 51764
      Source Pool :     A_POOL
      Elapsed Time :    09 seconds
      Application name :      ft
      Logical partition PE shape : 8 x 2 x 2
      Nodes in Partition :
```

```
        0x008 0x00a 0x00c 0x00e 0x018 0x01a 0x01c 0x01e
        0x108 0x10a 0x10c 0x10e 0x118 0x11a 0x11c 0x11e

   Partition 14:
      State : Active    Type :      Hardware
      Flags :
      Owner :     root (0)    Owning process: 51765
      Source Pool :     A_POOL
      Elapsed Time :    09 seconds
      Application name :      ft
      Logical partition PE shape : 8 x 2 x 2
      Nodes in Partition :
        0x020 0x022 0x024 0x026 0x030 0x032 0x034 0x036
        0x120 0x122 0x124 0x126 0x130 0x132 0x134 0x136


   Partition 16:
      State : Active    Type :      Hardware
      Flags :
      Owner :     root (0)    Owning process: 51766
      Source Pool :     A_POOL
      Elapsed Time :    09 seconds
      Application name :      ft
      Logical partition PE shape : 8 x 2 x 2
      Nodes in Partition :
        0x028 0x02a 0x02c 0x02e 0x038 0x03a 0x03c 0x03e
        0x128 0x12a 0x12c 0x12e 0x138 0x13a 0x13c 0x13e

Configuration Driver Statistics:

      Successful allocations: 7
            OS partitions: 0  HW partitions: 7
            Interactive allocations: 7

      Failed allocations: 0

      Active requests : 3 (high 5)

      Pending requests (normal priority) : 0 (high 6)

      Pending requests (high priority) : 0 (high 0)

Barrier bypass state summary:
      The following 33 barrier tree entries have circuit 0 flagged as in use:
            16    17    18    19    20    21    22    23
            32    33    34    35    36    37    38    39
            48    49    50    51    52    53    54    55
            70    71    74    75    78    79    83    84
            85
```

NAME

mppsysdmp – Dumps CRAY T3D system memory

SYNOPSIS

mppsysdmp [-g *gateway_device*] [-p *dump_dir_path*]

IMPLEMENTATION

Cray MPP systems

DESCRIPTION

The mppsysdmp utility captures areas of processing element (PE) control software memory. This information is useful when analyzing the CRAY T3D system after encountering uncertain or failed machine states. When a problem occurs, use the mppsysdmp utility to create a dump of the system memory.

The mppsysdmp utility initiates a partial system boot and dumps the memory. The memory data is dumped to a set of files (one binary file per PE in the system) in a dump directory within the specified directory (/core by default).

Each time you invoke mppsysdmp, a new dump directory is created within the specified or default directory.

At boot time, the mppstart utility creates a file in /mpp/cf/nodelist that contains the list of PEs that were booted in the system. This list is in turn used by the mppsysdmp utility to determine which PEs in the system should be dumped. If the /mpp/cf/nodelist file does not exist, a 32-PE system dump is generated by default.

For a 32-PE system with one I/O gateway, the mppsysdmp utility dumps about 10 Mwords of data, and completes in approximately 30 seconds. The size of the dump and the time to completion will increase linearly with the number of additional PEs in a system.

After the mppsysdmp utility completes, reboot the CRAY T3D system normally, using the mppstart(8) utility. The entire CRAY T3D system memory dump and subsequent reboot of a 32-PE system can be completed in one minute.

If a failure occurs on the mainframe serving as a frontend to the CRAY T3D system, the CRAY T3D system memory state remains intact. Following the dump and reboot of the frontend system, use the mppsysdmp utility to capture the CRAY T3D system memory state.

The mppsysdmp utility accepts the following options:

-g *gateway_device* Specifies the I/O node device through which the dump is to be accomplished. If this option is not specified, the mppsysdmp utility uses the I/O node device indicated in the DEFAULT_IOG environment variable. If the DEFAULOT_IOG environment variable is not set, the default I/O node device is /dev/mpp/iog01.

-p *dump_dir_path* Specifies the path to the directory within which the dump is to be accomplished. If this option is not specified, the mppsysdmp utility uses the path indicated in the DEFAULT_PREFIX environment variable. If the DEFAULOT_PREFIX environment variable is not set, the default path is /core.

NOTES

While the performance and size of CRAY T3D system memory dumps for smaller configurations is satisfactory, the current mppsysdmp utility contains only the first phase of the system memory dump implementation for the CRAY T3D system. The second phase includes optimizations through the use of data compression techniques and data redundancy reduction algorithms.

EXAMPLES

An example of executing the mppsysdmp utility and listing the files created is as follows:

```
% mppsysdmp
Cray T3D System Dump

Dumping Cray T3D system to /core/MPP.0503074934

% cd /core/MPP.0503074934
% ls -l
total 83968
-rw-r--r--  1 root      2621440 May  3 11:32 dump.0
-rw-r--r--  1 root      2621440 May  3 11:32 dump.1
-rw-r--r--  1 root      2621440 May  3 11:32 dump.10
-rw-r--r--  1 root      2621440 May  3 11:32 dump.100
-rw-r--r--  1 root      2621440 May  3 11:32 dump.101
-rw-r--r--  1 root      2621440 May  3 11:32 dump.102
-rw-r--r--  1 root      2621440 May  3 11:32 dump.103
-rw-r--r--  1 root      2621440 May  3 11:32 dump.11
-rw-r--r--  1 root      2621440 May  3 11:32 dump.110
-rw-r--r--  1 root      2621440 May  3 11:32 dump.111
-rw-r--r--  1 root      2621440 May  3 11:32 dump.112
-rw-r--r--  1 root      2621440 May  3 11:32 dump.113
-rw-r--r--  1 root      2621440 May  3 11:32 dump.12
-rw-r--r--  1 root      2621440 May  3 11:32 dump.120
-rw-r--r--  1 root      2621440 May  3 11:32 dump.121
-rw-r--r--  1 root      2621440 May  3 11:32 dump.122
-rw-r--r--  1 root      2621440 May  3 11:32 dump.123
-rw-r--r--  1 root      2621440 May  3 11:32 dump.13
-rw-r--r--  1 root      2621440 May  3 11:32 dump.130
-rw-r--r--  1 root      2621440 May  3 11:32 dump.131
-rw-r--r--  1 root      2621440 May  3 11:32 dump.132
-rw-r--r--  1 root      2621440 May  3 11:32 dump.133
-rw-r--r--  1 root      2621440 May  3 11:32 dump.2
-rw-r--r--  1 root      2621440 May  3 11:32 dump.20
-rw-r--r--  1 root      2621440 May  3 11:32 dump.21
-rw-r--r--  1 root      2621440 May  3 11:32 dump.22
-rw-r--r--  1 root      2621440 May  3 11:32 dump.23
-rw-r--r--  1 root      2621440 May  3 11:32 dump.3
-rw-r--r--  1 root      2621440 May  3 11:32 dump.30
-rw-r--r--  1 root      2621440 May  3 11:32 dump.31
-rw-r--r--  1 root      2621440 May  3 11:32 dump.32
-rw-r--r--  1 root      2621440 May  3 11:32 dump.33
-rw-r--r--  1 root      1048576 May  3 11:32 dump.4a0
-rw-r--r--  1 root      1048576 May  3 11:32 dump.4b0
```

FILES

`/core/MPP.`*mmddHHMMSS*`/dump.`*nnn*	Dump file created. Each file name includes the time of creation, which is shown as month, day, hour, minute, second (*mmddHHMMSS*), and the PE number (*nnn*).
`/mpp/cf/nodelist`	The list of PEs that were booted in the system.

SEE ALSO

`mppstart`(8)

NAME

olnx – Tests CRAY T3D interconnect network hardware

SYNOPSIS

/ce/bin/olnx [-cmb] [-s *start pass*] [-p *last pass*] [-t *hh:mm:ss*] [-u *hh:mm:ss*] [-a] [-E *pe list* | -D *pe list*] [-f *max*]

IMPLEMENTATION

CRAY T3D systems only

DESCRIPTION

The olnx command invokes the CRAY T3D interconnect network hardware confidence test. It is used to verify that a specified network partition is functional or to diagnose a failure in a partition that is suspected to have problems. Hardware faults that are not captured and reported by the microkernel will be captured and reported by the diagnostic. The olnx test builds a log of detected faults by printing information about each one to stdout. When the microkernel detects a hardware fault, the job is aborted. If this is the case, the test can be restarted repeatedly in order to build a log of failure information. The strategy is to isolate the fault based on a large set of failure information.

The olnx test has each PE perform random network operations using any destination PE and network packet type and command field. This tests the ability of the network logic to switch between directions, packet types, and contention levels. The program is event-driven so that operations can be overlapped (a message sent while a prefetch is pending, a processor-generated remote write performed while a block transfer is in progress, and so on).

The partition being tested can be any size.

Test enable options let you select the network operations to test. If no options are specified, the default is all options are enabled.

-c Enables processor-generated operations. These operations include cached and noncached normal reads, cached and noncached atomic swaps, fetch-and-increment reads and writes, cached read-ahead, single or cache-line writes, prefetch reads, prefetch atomic swap, and prefetch fetch and increment.

-m Enables hardware messages (using a memory function code of 7).

-b Enables block transfer operations.

Iteration control options let you control test iterations.

-s *start pass*
Starts olnx with the test vector generated during the pass specified by *start pass*.

-p *last pass*
Stops olnx at the pass specified by *last pass*.

-t *hh:mm:ss*
Runs olnx for the interval specified by *hh:mm:ss*. The test will terminate when this interval has expired. The default is to run forever.

-u *hh:mm:ss*
Writes the current pass count to stdout for the interval specified by *hh:mm:ss*. The default is no output of the pass count occurs.

PE select options allow you to enable or disable a subset of PEs in the partition. Only the PE's network operations are enabled or disabled. Do not use both the `-E` and the `-D` options at the same time. The default is all PEs are enabled. All local memories of the partition are used.

`-a` Causes address patterns to be used instead of random data.

[`-E` *pe list* | `-D` *pe list*]
Enables or disables individual PEs or ranges of PEs.

`-f` *max* Sets the maximum number of entries in the fault log printed to `stdout`. The default is 256.

EXAMPLES

Example 1: Runs all network operations forever.

```
olnx
```

Example 2: Runs processor-generated operations in PEs 0 through 13 and PE 37 for 1000 passes. The partition selected must be greater than or equal to 37 PEs.

```
olnx -c -E0-13,37 -p1000
```

Example 3: Runs all network operations in all PEs except PEs 12, 33, and 233 through 324 for 33,000 passes. The partition selected must be greater than or equal to 324 PEs.

```
olnx -D'33 12 233-324' -p33000
```

SEE ALSO

`olperi`(8) for information on the CRAY T3D processor chip user mode instructions test.

System Maintenance and Remote Testing Environment SMARTE) Guide, publication SPM–1017 (This manual is Cray Research Proprietary; dissemination of this documentation to non-CRI personnel requires approval from the appropriate vice president and a nondisclosure agreement. Export of technical information in this category may require a Letter of Assurance.)

CRAY T3D Diagnostic Reference Manual, publication CDM–0601–000 (This manual is Cray Research Proprietary; dissemination of this documentation to non-CRI personnel requires approval from the appropriate vice president and a nondisclosure agreement. Export of technical information in this category may require a Letter of Assurance.)

NAME

`olperi` – Tests CRAY T3D processor chip user mode instructions

SYNOPSIS

`/ce/bin/olperi [-fea] [-r] [-i] [-n` *instruction number*`] [-s` *start pass*`] [-p` *last pass*`] [-t` *hh*`:`*mm*`:`*ss*`] [-u` *hh*`:`*mm*`:`*ss*`]`

IMPLEMENTATION

CRAY T3D systems only

DESCRIPTION

The `olperi` (Processor Element Random Instruction diagnostic) command tests the processor chip user mode floating-point and integer instructions. It also tests the interface between the processor chip and the support logic. At each pass of the test, each PE builds an identical test vector, executes it, and then compares results between neighboring PEs. A test vector consists of a random instruction sequence and an initial data image. Any miscompare information is formatted and printed to `stdout`.

The results of execution are a trace buffer written as the random instructions are executed and the final state of the floating-point registers, integer registers, prefetch queue, DTB annex, fi register, and swaperand register.

When `olperi` detects a miscompare, the `-i` option enables the program to attempt to isolate the failure. It does this by removing instructions from the test vector one at a time and rerunning the test until the processors stop miscomparing.

The size of the partition being tested must be an even number of PEs, because each PE calculates its neighbor to be its own PE number exclusive OR-ed with 1. This keeps remote memory references used during the compare routine confined to a node, and likewise keeps failures that cause side effects confined to the node (for the most part).

Test enable options let you select the instructions to include in the test vector. If no options are specified, the default is all options are enabled.

`-f` Causes floating-point instructions to be included in the test vector.

`-e` Causes integer instructions to be included in the test vector.

`-a` Causes random cycle requests to be included in the test vector.

Iteration control options let you control test iterations.

`-r` Repeats the first test vector generated until a miscompare occurs. The default is a new test vector is generated at every pass.

`-i` Isolates the minimum failing test vector. The default is isolation is disabled.

`-n` *instruction number*
Specifies the number of machine instructions in the test vector. The value for *instruction number* can range from 16 through 1024.

`-s` *start pass*
Starts `olperi` with the test vector generated during the pass specified by *start pass*.

`-p` *last pass*
Stops `olperi` at the pass specified by *last pass*.

`-t` *hh*`:`*mm*`:`*ss*
Runs `olperi` for the interval specified by *hh*`:`*mm*`:`*ss*. The test will terminate when this interval has expired. The default is to run forever.

`-u` *hh:mm:ss*

Writes the current pass count to `stdout` for the interval specified by *hh:mm:ss*. The default is no output of the pass count occurs.

EXAMPLES

Example 1: Runs `olperi` starting with test vector 121 and repeats that test vector until a miscompare is detected.

```
olperi -s121 -r
```

Example 2: Runs random cycle requests only, enables the isolation option, and runs to pass 1000.

```
olperi -ai -p1000
```

Example 3: Runs test vectors that are 16 machine instructions long, includes only floating-point and integer instructions, starts at test vector 1000, and stops at test vector 2000.

```
olperi -n16 -fei -s1000 -p2000
```

SEE ALSO

`olnx`(8) for information on the CRAY T3D interconnect network hardware confidence test.

System Maintenance and Remote Testing Environment SMARTE) Guide, publication SPM–1017 (This manual is Cray Research Proprietary; dissemination of this documentation to non-CRI personnel requires approval from the appropriate vice president and a nondisclosure agreement. Export of technical information in this category may require a Letter of Assurance.)

CRAY T3D Diagnostic Reference Manual, publication CDM–0601–000 (This manual is Cray Research Proprietary; dissemination of this documentation to non-CRI personnel requires approval from the appropriate vice president and a nondisclosure agreement. Export of technical information in this category may require a Letter of Assurance.)

Index

Index

A

Activity monitoring, 21
Administrative resource pools, 1, 11, 15
    current usage, 24
    draining, 16
    marking UNAVAILABLE, 3
Applications (active), monitoring, 22
Attributes, administrative resource pool, 1
AVAILABLE pools, 3

B

BATCH pools, 2
Booting (rebooting), 18
BOTH pools, 2

C

Changing atrributes of administrative resource pools, 15
Choosing a shape for an administrative resource pool, 11
CRAY T3D administrative concepts, 1
CRAY T3D man pages, 6
CRAY T3D system messages, 31
CRAY T3D troubleshooting strategy, 27
Creating network routing tables, 4

D

Daemon log, MPP system (mppd.log), 5
Daemon log file, 28
Daemon, MPP system (mppd), 5
Determining the space needed for system dumps, 12
Dimension-order routing, 4
Draining administrative resource pools, 16
Dumping system memory, 29

E

Error messages, 31
Examining CRAY T3D system, 28
Express job pools, 3
Express processing, setting limits, 12

G

Group ID pools, 2

I

INTERACTIVE pools, 2

L

Limits, NQS queue, 25
Log files, 28
    monitoring, 21

M

Maintaining a CRAY T3D system, 15
Man pages, 6
Message log file, 28
Messages, system, 31
Modifying the UNICOS kernel tables, 11
Monitoring a CRAY T3D system, 21
Monitoring active CRAY T3D applications, 22
Monitoring CRAY T3D resources, 24
Monitoring CRAY T3D system activity, 21
Monitoring NQS status, 25
Monitoring PE status, 23
MPP daemon, 5

N

Network routing tables, 4
    creating, 4
    reconfiguring, 5
NQS limits, 25
NQS status, 25

P

Partitions, active, 24
PE status, 23
Performing a dump of CRAY T3D system memory, 29
Pools, administrative resoure, 1
Processing elements, monitoring status, 23

R

Rebooting the CRAY T3D system, 18
Reconfiguring network routing tables, 5
Resources, monitoring, 24
Routing
    dimension-order, 4
    network, 4
Routing table, 4
Routing tag, 4

S

Scheduling, 1
    express jobs, 3
Setting limits for express processing, 12
Shutting down the CRAY T3D system, 17
Startup (rebooting), 18
System dumps, reserving space, 12
System log file, 28
System message log file, 28
System messages, 31
System shutdown, 17
System startup, 18

T

Troubleshooting a CRAY T3D System, 27
Troubleshooting strategy, 27

U

UNAVAILABLE pools, 3
UNICOS kernel tables, 11
UNICOS man pages, 6
UNICOS parameter file, 10
Updating the UNICOS parameter file, 10

Reader's Comment Form

CRAY T3D Administrator's Guide SG–2507 1.1

Your reactions to this manual will help us provide you with better documentation. Please take a moment to complete the following items, and use the blank space for additional comments.

List the operating systems and programming languages you have used and the years of experience with each.

Your experience with Cray Research computer systems: _____0-1 year _____1-5 year _____5+years

How did you use this manual: _____in a class _____as a tutorial or introduction _____as a procedural guide _____as a reference _____for troubleshooting _____other

Please rate this manual on the following criteria:

	Excellent			Poor
Accuracy	4	3	2	1
Appropriateness (correct technical level)	4	3	2	1
Accessibility (ease of finding information)	4	3	2	1
Physical qualities (binding, printing, illustrations)	4	3	2	1
Terminology (correct, consistent, and clear)	4	3	2	1
Number of examples	4	3	2	1
Quality of examples	4	3	2	1
Index	4	3	2	1

Please use the space below for your comments about this manual. Please include general comments about the usefulness of this manual. If you have discovered inaccuracies or omissions, please specify the number of the page on which the problem occurred.

Name _____
Title _____
Company _____
Telephone _____
Today's date _____

Address _____
City _____
State/Country _____
Zip code _____
Electronic mail address _____

Cut along this line

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 6184 ST. PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY
RESEARCH, INC.

ATTN: Software Information Services
655 LONE OAK DR BLDG F
EAGAN MN 55121-9957

Fold

Reader's Comment Form

CRAY T3D Administrator's Guide SG–2507 1.1

Your reactions to this manual will help us provide you with better documentation. Please take a moment to complete the following items, and use the blank space for additional comments.

List the operating systems and programming languages you have used and the years of experience with each.

Your experience with Cray Research computer systems: _____0-1 year _____1-5 year _____5+years

How did you use this manual: _____in a class _____as a tutorial or introduction _____as a procedural guide _____as a reference _____for troubleshooting _____other

Please rate this manual on the following criteria:

	Excellent			Poor
Accuracy	4	3	2	1
Appropriateness (correct technical level)	4	3	2	1
Accessibility (ease of finding information)	4	3	2	1
Physical qualities (binding, printing, illustrations)	4	3	2	1
Terminology (correct, consistent, and clear)	4	3	2	1
Number of examples	4	3	2	1
Quality of examples	4	3	2	1
Index	4	3	2	1

Please use the space below for your comments about this manual. Please include general comments about the usefulness of this manual. If you have discovered inaccuracies or omissions, please specify the number of the page on which the problem occurred.

Name _____
Title _____
Company _____
Telephone _____
Today's date _____

Address _____
City _____
State/Country _____
Zip code _____
Electronic mail address _____

Cut along this line

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 6184 ST. PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY
RESEARCH, INC.

ATTN: Software Information Services
655 LONE OAK DR BLDG F
EAGAN MN 55121-9957

Fold